ISSN : 0294-6068

pom's

5

Sommaire

Editorial	3
Le clavier magique	4
Bases de données sur Apple	11
La programmation facilitée	22
Banc-test de la carte Legend 128 K DE	33
HAIFA : un amper-interpréteur complet	35
Création graphique en Pascal	54
CX Multigestion à l'essai	61
Courrier des lecteurs	64
Bibliographie	66

NUMÉRO 5 - SEPTEMBRE 1982 - 35 Francs

JCR, DES MICRO-ORDINATEURS PROFESSIONNEL ET GRAND PUBLIC.

JEUX
ÉDUCATION
GESTION
BUREAUTIQUE

APPLE II

PROMOTION Nous consulter.

Le plus populaire des micro-ordinateurs. 48 K RAM. Basic Applesoft. Une gamme incomparable de logiciels et d'accessoires.

Apple II + 48 K + Disk avec Contrôleur + Moniteur 12".

SHARP PC 1500

Ordinateur de poche de 1,5 Ko de mémoire vive extensible avec module de 8 K CE 155.

CE 150

Mini table traçante 4 couleurs directement connectable sur PC 1500. Interface K 7 incorporé.

PC 1500 + CE 150. **4 100 F**

CE 158

EPSON HX 20

Un système compact clavier écran LCD avec imprimante. **5 900 F**

Micro K 7. **1 200 F**

Extension 16 K. **1 300 F**

EPSON

Imprimantes de haute qualité d'impression. Interface parallèle type Centronics.

MX 80 FT : 80 cps. ou 132 compressés. **5 800 F**

MX 100 : 100 cps. 132 caractères ou 233 compressés. **8 200 F**

APPLE III

L'outil professionnel par excellence. 128 Ko ou 256 Ko. Unité de disque incorporée. Sortie RS 232. Nombreux interfaces disponibles. Adjonction possible d'un disque dur de 5 méga. Profilé. Écran vert haute résolution antireflets. Clavier Azerty - Qwerty. Nous consulter.

COMMODORE VIC 20

Un vrai micro-ordinateur puissant et évolutif idéal pour l'initiation comme pour la pratique de la programmation. 16 couleurs RAM 3,5 K. Version en PAL. **2 350 F**

NOUVEAU CHEZ JCR

- ATARI 400 et 800
- APPLE II E
- CASIO PB 100
- SHARP PC 1251
- SHARP PC 1212
- INTERF. RS 232/PC 1500
- VICTOR II 48 K HR

TO 7 THOMSON

Un ordinateur 100% français 8 Ko extensible à 32 Ko. Fourni avec un lecteur optique. Sortie couleur Péritel. Clavier Azerty accentué. Idéal pour apprendre en famille. **3 650 F**

Vente par correspondance
Catalogue gratuit sur demande
Crédit 4-36 mois
Leasing 36-48 mois

En raison des fluctuations monétaires, ces prix sont susceptibles d'être modifiés sans préavis. Nous consulter pour confirmation.

BOUTIQUE

58, rue Notre-Dame-de-Lorette
75009 PARIS
Tél. (1) 282.19.80 – Télex : 290350 F

59, rue du Docteur Escat
13006 MARSEILLE
Tél. (91) 37.62.33

Horaires d'ouverture du magasin - du mardi au samedi : 10 h - 12 h 45 / 14 h - 19 h.

Editorial

L'année scolaire qui débute sera l'occasion de développements importants dans le domaine de la micro-informatique. Nous en avons déjà parlé dans le dernier éditorial, mais nous devrions voir apparaître en septembre les premiers matériels IBM PC et de la série Digital réellement disponibles à la vente. Les Sirius devraient voir leur logiciel s'enrichir. Pour nous, utilisateurs de matériels Apple, ITT, Silex et autres, les nouveautés sont les périphériques, le développement de la communication, des réseaux de toutes sortes et, bien sûr, l'apparition de programmes toujours plus variés et aux possibilités multiples.

Nous entamons avec ce cinquième numéro de Pom's la deuxième année de notre revue. La première année s'est passée de façon satisfaisante : nous sommes maintenant quasiment en rupture de stock pour chacun des quatre premiers numéros, et nous venons d'augmenter le tirage de Pom's, passant ainsi de 3 000 à 4 000 exemplaires par numéro. La seule chose qui a freiné notre développement a été le refus de la part de l'administration française de nous reconnaître comme revue de presse, en nous accordant l'accord de la Commission Paritaire des Publications de Presse. On reproche à Pom's d'être, soi-disant, une revue publicitaire pour Apple ; la conséquence principale étant que nos frais sont ainsi majorés, par le jeu de la TVA et des frais postaux, de 30 %.

Nous continuons par conséquent à travailler avec des moyens relativement simples, mais faisons tout notre possible pour vous offrir une revue de qualité, sinon de présentation totalement professionnelle. A en croire la croissance des ventes de Pom's, le temps énorme que nous consacrons à Pom's n'est pas perdu. Nous en sommes heureux et espérons que Pom's continuera d'être pour vous un moyen de communication, d'apprentissage et de découverte.

Nous vous offrons dans ce numéro une contribution qui est en fait, vu sa richesse, un produit valant largement de nombreux logiciels commerciaux. HAIFA, le Herz Amper Interpreter For Applesoft, étendra les possibilités de programmation de votre Apple, en particulier dans les domaines graphique et sonore. Le programme est tellement important que nous n'avons pu mettre le source (Lisa 2.5) sur la disquette d'accompagnement. Ce code source occupant plus de 150 secteurs, nous n'avons pu non plus le lister dans la revue : nous imaginons mal que quelqu'un tienne absolument à rentrer plus de quatre pages pleines, ne serait-ce que pour le code hexa ! La disquette comporte par conséquent la version binaire de HAIFA, et des programmes de démonstration, sous forme de générique pour Pom's. Les codes sources des versions 48K et 64K pourront être commandés sur une disquette spéciale vendue en option, au prix habituel de 50 francs TTC.

Nous vous offrons aussi une analyse des bases de données réalisée par J.-F. Duvivier, un clavier magique et un amper-interpréteur (plus simple que celui de Herz) dûs à Denis Sureau, un banc-test du programme CX-Multigestion et de la carte Legend 128K DE. Les fans de Pascal trouveront encore dans cet exemplaire une excellente contribution de Michel Crimont, qui s'attaque ici à la création de formes.

Nous recevons de plus en plus de contributions de lecteurs, et n'avons d'ailleurs pas pu toutes les intégrer dans cet exemplaire. Nous rappelons aux lecteurs qu'ils peuvent nous envoyer leurs contributions sur disquette. Quand cela est possible, nous préférons recevoir le texte sur la disquette en format Apple Writer I ou II.

Directeur de la publication — rédacteur en chef : Hervé Thiriez — *Siège social :* Editions MEV — 49, rue Lamartine — 78000 Versailles — *Rédaction et abonnements :* 59, bd de Glatigny — 78000 Versailles — Tél. (3) 918.13.07 — *Régie publicitaire :* Force 7 — 41, rue de la Grange-aux-Belles — 75483 Paris Cedex 10 — Tél. (1) 238.66.10 — *Diffusion auprès des boutiques informatiques et librairies :* Editions du PSI — 41-51, rue Jacquard — B.P. 86 — Tél. (6) 007.59.31.

Ont collaboré à ce numéro : Christian Colmant, Michel Crimont, Jean-François Duvivier, Olivier Herz, Denis Sureau, Hervé Thiriez — *Conseillers techniques :* Olivier Herz et Gérard Michel — *Dessins :* Bertrand Delestrée et Jean Mourot.

Le clavier magique

Denis Sureau

Ce programme utilitaire permet de donner à l'Apple une possibilité que l'on trouve d'origine sur le PET et qui facilite grandement l'entrée d'un programme BASIC au clavier : une seule lettre est tapée, associée à la touche CTRL et une instruction Applesoft apparait sur l'écran : ainsi, CTRL-I remplacera INPUT, CTRL-T THEN, etc...

Pour que la frappe d'un programme, tâche fastidieuse s'il en est, soit encore facilitée, l'autonumérotation est permise par le programme: il suffit d'appuyer sur la barre d'espacement pour que le numéro de ligne suivant apparaisse avec à sa droite le curseur...

Le listing du programme donne toutes les indications nécessaires sur son fonctionnement; il est relativement court car il se sert de la table des instructions Applesoft, commençant à partir de l'adresse $D0D0, grâce à une table d'adresses donnant la correspondance entre les lettres A à Z et 22 instructions Applesoft. Les touches CTRL-U, CTRL-H, CTRL-M et CTRL-X ne sont pas affectées, CTRL-M équivaut à RETURN pour le moniteur.

Lorsqu'on le charge, le programme est inactif. On le rend actif par CTRL-SHIFT-P et l'on peut ensuite, même à l'intérieur d'une ligne, le déconnecter ou le reconnecter avec cette commande.

Quand au numéro de départ de la numérotation automatique et à l'incrément, ils sont définis par l'instruction & [DEPART],[INCREMENT]. Par défaut, le numéro de départ est 100 et l'incrément 10.

On pourra également obtenir la commande LIST avec la touche "/" et CATALOG avec la touche "$", mais seulement en exécution directe.

La correspondance des touches de contrôle est la suivante :

Touche	Equivalent	Touche	Equivalent
A	INVERSE	O	HPLOT
B	HTAB	P	HOME
C	COLOR	Q	POKE
D	DATA	R	RETURN
E	TEXT	S	SCALE
F	FOR	T	THEN
G	GOTO	V	VTAB
I	INPUT	W	PEEK
J	GOSUB	Y	XDRAW
K	VLIN	Z	NORMAL
L	HLIN	CTRL SHIFT M	XDRAW
N	NEXT		

Bascule pour brancher ou débrancher le clavier magique : CTRL-SHIFT-P.

```
10  TEXT : HOME
20  HTAB 10: PRINT "LE CLAVIER MA
     GIQUE"
30  PRINT : PRINT
40  PRINT " CET UTILITAIRE EN ASS
     EMBLEUR PERMET"
50  PRINT "D'ECRIRE LES INSTRUCTI
     ONS APPLESOFT AU"
60  PRINT "MOYEN DE DEUX TOUCHES:
     "
70  PRINT " - LA TOUCHE CTRL
80  PRINT " - ASSOCIEE A UN CARAC
     TERE DE A A Z"
90  PRINT "   NORMALEMENT L'INITI
     ALE DE L'INS-     ": PRINT "
       TRUCTION."
100 PRINT "    DE PLUS ON OBTIEN
     T AUTOMATIQUEMENT"
110 PRINT "LE NUMERO DE LA LIGNE
      SUIVANTE EN"
120 PRINT "APPUYANT SUR LA BARRE
      D'ESPACE."
130 PRINT "   LE NUMERO DE DEPAR
     T DE L'AUTONUMERO-": PRINT "
     TATION ET L'INCREMENT SONT D
     ONNES"
140 PRINT "PAR L'INSTRUCTION: ";
     : INVERSE : PRINT "&<DEP>,<I
     NC>": NORMAL
150 PRINT "    ON MET LE PROGRAM
     ME EN FONCTION ET"
160 PRINT "ON PEUT -A TOUT MOMEN
     T- LE METTRE HORS"
170 PRINT "FONCTION PAR LA COMMA
     NDE:"
180 HTAB 15: INVERSE : PRINT " C
     TRL-S ": NORMAL
190 IF  PEEK (115) +  PEEK (116)
      * 256 < 22000 THEN 300
200 PRINT  CHR$ (4)"BRUN  MAGIQ$
     6000"
210 CALL 24576
220 END
300 PRINT  CHR$ (4)"BRUN MAGIQ$3
     A00"
310 CALL 14848
320 END
```

```
0800            1  ;&&&&&&&&&&&&&&&&&&&&
0800            2  ;&                  &
0800            3  ;&       L E        &
0800            4  ;&                  &
0800            5  ;&  C L A V I E R   &
0800            6  ;&                  &
0800            7  ;&  M A G I Q U E   &
0800            8  ;&                  &
0800            9  ;&       PAR        &
0800           10  ;&   DENIS SUREAU   &
0800           11  ;&                  &
0800           12  ;&&&&&&&&&&&&&&&&&&&&
0800           13  ;
0800           14  HOOK   EPZ $38
0800           15  KEYIN  EQU $FD1B
0800           16  PRINT  EQU $FDED
0800           17  INLINZ EQU $D52E
0800           18  FRMNUM EQU $DD67
0800           19  LINNUM EPZ $50
0800           20  CHRGET EPZ $B1
0800           21  TXTPTR EPZ $B8
0800           22  GETADR EQU $E752
0800           23  INCH   EPZ $18
0800           24  INCL   EPZ $19
0800           25  NUML   EPZ $1A
0800           26  NUMH   EPZ $1B
0800           27  PASSE  EPZ $1C              ;0=OFF,1=ON
0800           28  LINPTR EQU $ED24
0800           29  BELL2  EQU $FBE4
0800           30  WAIT   EQU $FCA8
0800           31  ;
6000           32         ORG $6000
6000           33         OBJ $800
6000           34  ;
6000           35  ; (LONGUEUR:$1CE)
6000           36  ;
6000           37  INIT:
6000 A923      38         LDA #TESTS           ;PLACE UN
6002 8538      39         STA HOOK             ;SAUT A
6004 A960      40         LDA /TESTS           ;L'ADRESSE TESTS
6006 8539      41         STA HOOK+1           ;DANS LE 'INPUT HOOK'
6008 A900      42         LDA #$00
600A 851B      43         STA NUMH
600C 8518      44         STA INCH
600E 851C      45         STA PASSE
6010 A964      46         LDA #$64
6012 851A      47         STA NUML
6014 A90A      48         LDA #$0A
6016 8519      49         STA INCL
6018 A900      50         LDA #INIT
601A 8573      51         STA $73
601C A960      52         LDA /INIT
601E 8574      53         STA $74
6020 4CEA03    54         JMP $3EA
6023           55  TESTS:
6023 201BFD    56         JSR KEYIN            ;LECTURE CLAVIER
6026 C980      57         CMP #$80             ;CTRL-@?
6028 D003      58         BNE TST1
602A 4CE860    59         JMP MAGIQ
602D 48        60  TST1   PHA
```

```
602E A51C      61          LDA PASSE          ;ON-OFF?
6030 D002      62          BNE TST2
6032 68        63          PLA
6033 60        64          RTS
6034 68        65  TST2    PLA
6035 C99E      66          CMP #$9E           ;TEST CLAVIER
6037 901B      67          BCC CTRL           ;TOUCHE DE CONTROLE?
6039 E000      68          CPX #$00
603B D016      69          BNE RTS
603D C9A0      70          CMP #$A0           ;BARRE D'ESPACE?
603F F050      71          BEQ NUM
6041 C9A3      72          CMP #"#"
6043 F073      73          BEQ PARAM
6045 C9A4      74          CMP #"$"
6047 D003      75          BNE TST3
6049 4CA261    76          JMP CATALG
604C C9AF      77  TST3    CMP #"/"
604E D003      78          BNE RTS
6050 4CAA61    79          JMP LIST
6053 60        80  RTS     RTS
6054           81  CTRL:
6054 C98D      82          CMP #$8D           ;RETURN?
6056 F0FB      83          BEQ RTS
6058 C988      84          CMP #$88           ;FLECHE GAUCHE?
605A F0F7      85          BEQ RTS
605C C995      86          CMP #$95           ;FLECHE DROITE?
605E F0F3      87          BEQ RTS
6060 C998      88          CMP #$98           ;CTRL-X?
6062 F0EF      89          BEQ RTS
6064 C99B      90          CMP #$9B           ;ESC
6066 F0EB      91          BEQ RTS
6068           92  ;
6068           93  ; CETTE PROCEDURE SELECTIONNE
6068           94  ;DANS LA TABLE DU PROGRAMME
6068           95  ;L'ADRESSE DE L'INSTRUCTION
6068           96  ;VOULUE DANS LA TABLE DE
6068           97  ;L'APPLESOFT.
6068           98  ;
6068 D8        99          CLD
6069 297F     100          AND #$7F           ;A=A-128
606B 0A       101          ASL                ;A=A*2
606C A8       102          TAY
606D B94F61   103          LDA TABLE,Y
6070 F01E     104          BEQ EXIT           ;SI 0000,SORTIE
6072 8D7F60   105          STA ADRESS+2
6075 B95061   106          LDA TABLE+1,Y
6078 8D7E60   107          STA ADRESS+1
607B          108  INSTR:
607B A000     109          LDY #$00
607D B9FFFF   110  ADRESS  LDA $FFFF,Y
6080 C980     111          CMP #$80           ;DERNIER CARACTERE?
6082 100C     112          BPL EXIT
6084          113  ;AFFICHAGE ET INSERTION DE CHAQUE
6084          114  ;CARACTERE DANS LE BUFFER D'ENTREE
6084 0980     115          ORA #$80
6086 20EDFD   116          JSR PRINT
6089 9D0002   117          STA $200,X
608C E8       118          INX
608D C8       119          INY
608E D0ED     120          BNE ADRESS
```

```
6090 60        121  EXIT   RTS
6091           122  ;
6091           123  ;--------------------
6091           124  ;    A U T O N U M
6091           125  ;--------------------
6091           126  ;AFFICHE LE NUMERO DE LIGNE
6091           127  NUM:
6091 A61A      128         LDX NUML
6093 A51B      129         LDA NUMH
6095 2024ED    130         JSR LINPTR
6098 A001      131         LDY #01
609A           132  ;INSERTION DANS LE BUFFER D'ENTREE
609A           133  NUM2:
609A B128      134         LDA ($28),Y
609C C9A0      135         CMP #$A0
609E F007      136         BEQ NUM3
60A0 9D0002    137         STA $200,X
60A3 C8        138         INY
60A4 E8        139         INX
60A5 D0F3      140         BNE NUM2
60A7           141  ;
60A7           142  ;INCREMENT DU NUMERO SUIVANT
60A7           143  NUM3:
60A7 D8        144         CLD
60A8 18        145         CLC
60A9 A519      146         LDA INCL
60AB 651A      147         ADC NUML
60AD 851A      148         STA NUML
60AF A518      149         LDA INCH
60B1 651B      150         ADC NUMH
60B3 851B      151         STA NUMH
60B5 A9A0      152         LDA #$A0
60B7 60        153         RTS
60B8           154  ;
60B8           155  ;ENTREE DES NUMERO DE DEPART & INC.
60B8           156  PARAM:
60B8 AA        157         TAX
60B9 C624      158         DEC $24
60BB 202ED5    159         JSR INLIN2          ;ENTREE
60BE A900      160         LDA #$00            ;D'UNE LIGNE
60C0 85B8      161         STA TXTPTR
60C2 A902      162         LDA #$02
60C4 85B9      163         STA TXTPTR+1
60C6 2067DD    164         JSR FRMNUM          ;EVALUATION
60C9 2052E7    165         JSR GETADR
60CC A551      166         LDA LINNUM+1
60CE A450      167         LDY LINNUM
60D0 851B      168         STA NUMH
60D2 841A      169         STY NUML
60D4 20B100    170         JSR CHRGET
60D7 2067DD    171         JSR FRMNUM
60DA 2052E7    172         JSR GETADR
60DD A450      173         LDY LINNUM
60DF A551      174         LDA LINNUM+1
60E1 8518      175         STA INCH
60E3 8419      176         STY INCL
60E5 A980      177         LDA #$8D
60E7 60        178         RTS
60E8           179  ;
60E8           180  ; ROUTINES D'AFFICHAGE DU TITRE
```

```
60EB 8606     181  MAGIQ  STX $6
60EA A524     182         LDA $24          ;CONSERVATION
60EC 8508     183         STA $8           ;DES VTAB
60EE A525     184         LDA $25          ;ET HTAB
60F0 8509     185         STA $9           ;INITIAUX
60F2 A51C     186         LDA PASSE
60F4 D048     187         BNE EFFACE
60F6          188  TITRE:
60F6 8606     189         STX $6
60F8 A900     190         LDA #00
60FA 8525     191         STA $25
60FC 8524     192         STA $24
60FE 2022FC   193         JSR $FC22
6101 A20B     194         LDX #$0B
6103 204AF9   195         JSR $F94A
6106 A000     196         LDY #00
6108 B98B61   197  T2     LDA T,Y
610B C8       198         INY
610C 20EDFD   199         JSR PRINT
610F C98D     200         CMP #$8D
6111 D0F5     201         BNE T2
6113 A606     202         LDX $6
6115 A901     203         LDA #$1
6117 8522     204  RTSTT  STA $22
6119 851C     205         STA PASSE
611B A508     206         LDA $8
611D 8524     207         STA $24
611F A509     208         LDA $9
6121 8525     209         STA $25
6123 2022FC   210         JSR $FC22
6126 A020     211         LDY #$20
6128 20E4FB   212         JSR BELL2
612B A9D0     213         LDA #$D0
612D 20A8FC   214         JSR WAIT
6130 A018     215         LDY #$18
6132 20E4FB   216         JSR BELL2
6135 A9A0     217         LDA #$A0
6137 A606     218         LDX $6
6139 D002     219         BNE RTSTT2
613B A98D     220         LDA #$8D
613D 60       221  RTSTT2 RTS
613E          222  EFFACE:
613E A000     223         LDY #$00
6140 8424     224         STY $24
6142 8425     225         STY $25
6144 2022FC   226         JSR $FC22
6147 209CFC   227         JSR $FC9C
614A A900     228         LDA #$00
614C 4C1761   229         JMP RTSTT
614F          230  TABLE:
614F 0000D1   231         HEX 0000D14FD125D110
6152 4FD125
6155 D110
6157 D0DAD0   232         HEX D0DAD0EFD0D3D193
615A EFD0D3
615D D193
615F 0000D0   233         HEX 0000D0DED1A4D105
6162 DED1A4
6165 D105
6167 D10100   234         HEX D1010000D0D6D117
```

```
616A 00D0D6
616D D117
616F D129D1  235          HEX D129D1C7D1A9D131
6172 C7D1A9
6175 D131
6177 D1EF00  236          HEX D1EF0000D164D23B
617A 00D164
617D D23B
617F 0000D1  237          HEX 0000D11CD1490000
6182 1CD149
6185 0000
6187 0000D1  238          HEX 0000D120
618A 20
618B 3A200C  239  T       INV ": LE CLAVIER MAGIQUE :"
618E 052003
6191 0C0116
6194 090512
6197 200D01
619A 070911
619D 150520
61A0 3A
61A1 8D      240          HEX 8D
61A2         241  ;
61A2         242  ; TOUTE COMMANDE PEUT ETRE
61A2         243  ; AJOUTEE AVEC LA ROUTINE
61A2         244  ; 'COMM' EN AJOUTANT CETTE
61A2         245  ; COMMANDE A LA LISTE 'LST'
61A2         246  ; ON DONNE A Y LA PLACE DU
61A2         247  ; PREMIER CARACTERE DE LA
61A2         248  ; COMMANDE ET ON BRANCHE SUR
61A2         249  ; COMM  (Y DE 0 A Z54)
61A2         250  ; ON AJOUTE UN TEST EN DEBUT
61A2         251  ; DU PROGRAMME.
61A2         252  ; COMME AVEC CATALOG CI-DESSOUS:
61A2         253  ;
61A2         254  ;-------------------
61A2         255  ;  C A T A L O G
61A2         256  ;-------------------
61A2 A005    257  CATALG LDY #$05
61A4 20AC61  258         JSR COMM
61A7 A9BD    259         LDA #$BD
61A9 60      260         RTS
61AA         261  ;-------------------
61AA         262  ;    L I S T
61AA         263  ;-------------------
61AA A000    264  LIST   LDY #$00
61AC A200    265  COMM   LDX #$00
61AE B9C061  266  LLOOP  LDA LST,Y
61B1 D003    267         BNE LLOOP2
61B3 A9A0    268         LDA #$A0
61B5 60      269         RTS
61B6 20EDFD  270  LLOOP2 JSR PRINT
61B9 9D0002  271         STA $200,X
61BC E8      272         INX
61BD C8      273         INY
61BE D0EE    274         BNE LLOOP
61C0         275  ;
61C0 CCC9D3  276  LST    ASC "LIST"
61C3 D4
61C4 00      277         HEX 00
61C5 C3C1D4  278         ASC "CATALOG"
61C8 C1CCCF
61CB C7
61CC 00      279         HEX 00
             280  FIN    END
```

DE L'INTELLIGENCE EN MEMOIRE...

apple ITT2020

CARTE M/DOS 6502

LE SYSTEME D'EXPLOITATION DU 6502 - MONOPOSTE/MULTIPOSTE

- POUR UNE PROGRAMMATION SIMPLIFIEE
- POUR DIVISER PAR 20 LA LONGUEUR DE VOS PROGRAMMES
- POUR GERER DES MEMOIRES DE 140 K A 120 MEGAS
- POUR GERER VOS FICHIERS SEQUENTIELS INDEXES MULTICLES
- POUR GERER VOTRE ECRAN PAR MASQUES DE SAISIE (ADAPTABLE AUX CARTES 80 COLONNES)
- POUR GERER VOTRE IMPRIMANTE PAR MASQUES D'IMPRESSION

LA VERSION MULTIPOSTE VOUS ASSURE :
- LA MISE EN COMMUN TOTALE DES RESSOURCES SANS CONFLIT
- AUTONOMIE DES POSTES INTELLIGENTS DISPOSANT DE LEUR PROPRE UNITE CENTRALE

DISTRIBUTEURS AGREES :

MICRO ALPHA SOFT
11, Impasse du Lacquet
25200 MONTBELIARD
Tél. (81) 97.16.46

MICROMEGAS
22, rue des 3 Pierres
69007 LYON
Tél. (7) 861.19.52

D.S.A. INFORMATIQUE
5, bd Dubouchage
06000 NICE
Tél. (93) 85.15.96

SEEMI
61, rue Ch. Rivière - B.P. 0701
44401 REZE CEDEX
Tél. (40) 75.52.80

S.E.P. Marketing

UN PRODUIT micro informatique service

3, rue Meyerbeer, 06000 NICE, tél. (93) 87.74.67

Récapitulation 48K

```
6000- A9 23 85 38 A9 60 85 39
6008- A9 00 85 1B 85 18 85 1C
6010- A9 64 85 1A A9 0A 85 19
6018- A9 00 85 73 A9 60 85 74
6020- 4C EA 03 20 1B FD C9 80
6028- D0 03 4C E8 60 48 A5 1C
6030- D0 02 68 60 68 C9 9E 90
6038- 1B E0 00 D0 16 C9 A0 F0
6040- 50 C9 A3 F0 73 C9 A4 D0
6048- 03 4C A2 61 C9 AF D0 03
6050- 4C AA 61 60 C9 8D F0 FB
6058- C9 88 F0 F7 C9 95 F0 F3
6060- C9 98 F0 EF C9 9B F0 EB
6068- D8 29 7F 0A A8 B9 4F 61
6070- F0 1E 8D 7F 60 B9 50 61
6078- 8D 7E 60 A0 00 B9 FF FF
6080- C9 80 10 0C 09 80 20 ED
6088- FD 9D 00 02 E8 C8 D0 ED
6090- 60 A6 1A A5 1B 20 24 ED
6098- A0 01 B1 28 C9 A0 F0 07
60A0- 9D 00 02 C8 E8 D0 F3 D8
60A8- 18 A5 19 65 1A 85 1A A5
60B0- 18 65 1B 85 1B A9 A0 60
60B8- AA C6 24 20 2E D5 A9 00
60C0- 85 B8 A9 02 85 B9 20 67
60C8- DD 20 52 E7 A5 51 A4 50
60D0- 85 1B 84 1A 20 B1 00 20
60D8- 67 DD 20 52 E7 A4 50 A5
60E0- 51 85 18 84 19 A9 8D 60
60E8- 86 06 A5 24 85 08 A5 25
60F0- 85 09 A5 1C D0 48 86 06
60F8- A9 00 85 25 85 24 20 22
6100- FC A2 0B 20 4A F9 A0 00
6108- B9 8B 61 C8 20 ED FD C9
6110- 8D D0 F5 A6 06 A9 01 85
6118- 22 85 1C A5 08 85 24 A5
6120- 09 85 25 20 22 FC A0 20
6128- 20 E4 FB A9 D0 20 A8 FC
6130- A0 18 20 E4 FB A9 A0 A6
6138- 06 D0 02 A9 8D 60 A0 00
6140- 84 24 84 25 20 22 FC 20
6148- 9C FC A9 00 4C 17 61 00
6150- 00 D1 4F D1 25 D1 10 D0
6158- DA D0 EF D0 D3 D1 93 00
6160- 00 D0 DE D1 A4 D1 05 D1
6168- 01 00 00 D0 D6 D1 17 D1
6170- 29 D1 C7 D1 A9 D1 31 D1
6178- EF 00 00 D1 64 D2 3B 00
6180- 00 D1 1C D1 49 00 00 00
6188- 00 D1 20 3A 20 0C 05 20
6190- 03 0C 01 16 09 05 12 20
6198- 0D 01 07 09 11 15 05 20
61A0- 3A 8D A0 05 20 AC 61 A9
61A8- 8D 60 A0 00 A2 00 B9 C0
61B0- 61 D0 03 A9 A0 60 20 ED
61B8- FD 9D 00 02 E8 C8 D0 EE
61C0- CC C9 D3 D4 00 C3 C1 D4
61C8- C1 CC CF C7 00
```

Bases de données sur Apple

Jean-François Duvivier

Il est généralement admis que l'une des utilisations les plus populaires d'un ordinateur individuel consiste à stocker des données et à pouvoir les retrouver rapidement. Cela se confirme au vu de l'importance et du nombre de programmes de gestion de bases de données (ou gestion de fichiers) disponibles sur le marché. Au même titre que les jeux ou les progiciels de traitement de texte, ils constituent une classe tout à fait particulière de programmes.

Leur première apparition coïncide avec l'arrivée sur l'Apple des unités de disquettes (DOS 3.1 en juillet 1978, DOS 3.2 en février 1979), tant il est vrai qu'une gestion de fichier sur cassette était problématique. Ces premiers essais ont été l'oeuvre de clubs naissants, tel que Call-APPLE. Puis, avec l'apparition des SSCI attirées par le succès et la diffusion d'Apple, les programmes commerciaux se sont multipliés. Avec la concurrence, la qualité de ces progiciels s'est améliorée rapidement pour déboucher sur des produits tout à fait remarquables.

J'ai choisi de vous en présenter quatre parmi les meilleurs. Vous n'y trouverez peut-être pas votre favori, le seul qui trouve grâce à vos yeux, et je vous prie de m'excuser d'avance pour cet oubli ! A chacun de ces quatre progiciels sont consacrées deux à trois pages qui vous permettront de vous faire une idée plus précise de leur fonctionnement et de leurs possibilités. J'ai respecté autant que possible le même cheminement tout au long de cette analyse, avec des étapes bien définies, afin que vous puissiez plus facilement effectuer des comparaisons. ux. Enfin, à la suite de ces textes, un tableau récapitulatif rassemble les principales caractéristiques de ces progiciels.

Vous ne trouverez pas à la fin de cet article une conclusion générale, car il était impossible de répondre à la question : "Quel est le meilleur ?" ou "Lequel dois-je choisir ?". Chaque programme a ses avantages et ses inconvénients. Le choix dépend surtout de l'utilisation qui doit en être faite. En guise d'aide, vous pourrez consulter l'article paru dans l'Ordinateur Individuel d'avril 1982 : "Quelques conseils pour bien choisir un progiciel de gestion de fichier", article dont je recommande la lecture préalable à tous ceux qui n'ont jamais utilisé de programmes de gestion de fichiers et qui pourraient être déroutés par les termes employés dans cet article.

Quelques mots avant de commencer, sur les absents. Il s'agit notamment de :
- "CX Multigestion", produit français, qui fait l'objet d'une analyse détaillée par Hervé Thiriez dans ce numéro.
- "Data Factory" écrit par Bill Passauer et distribué par Micro-Lab (150 $ aux USA). Ce produit monte très fort dans les classements mensuels publiés par Softalk, mais je n'ai pas pu me le procurer à temps.
- "Modifiable Database" (Synergistic Software), programme qui a quelques défenseurs acharnés car ses fichiers sont totalement accessibles et utilisables en Basic, et à cause de sa grande facilité d'utilisation.
- "Super Généfiche" (Imagol) dont nous attendons de pouvoir tester la version professionnelle.
- des progiciels exécutés sous CP/M avec la carte Z80 (Softcard). Il faut surtout citer "Datastar" (la star des programmes de gestion de fichiers sous CP/M) et "Unibase". Bien qu'ils possèdent des qualités évidentes et indiscutées, je ne les ai pas fait figurer ici pour ne pas rompre "l'homogénéité matérielle", l'Apple II équipé d'une Softcard n'étant plus tout à fait un Apple II...

En dernier lieu, j'adresse tous mes remerciements à la société SIDEC qui m'a prêté très amicalement les programmes faisant l'objet de cette étude.

CCA DATA MANAGEMENT SYSTEM : UN ANCIEN QUI A DE BEAUX RESTES

CCA DMS est le plus ancien des progiciels présentés ici puisque sa première apparition remonte à 1979, date à laquelle le DOS 3.3 n'existait pas encore ! Distribué par Personal Software (si,si ! vous connaissez : ils distribuent aussi Visicalc), ses possibilités lui ont vite permis d'acquérir une place de choix parmi les programmes de l'époque. Présenté sur une disquette normale, c'est-à-dire non protégée (1979, c'était la belle époque), CCA a connu un vif succès et une distribution (...) très large qui lui a permis de se retrouver en bonne place dans la programmathèque de nombreux utilisateurs Apple.

L'adaptation au DOS 3.3, Muffin oblige, n'a

pas présenté de difficultés. Ayant ainsi traversé le temps sans encombres, le CCA DMS se retrouve parmi les programmes de gestion de fichiers les plus connus, malgré des performances en retrait par rapport aux progiciels plus récents.

Configuration

CCA DMS se présente sous forme de six programmes Applesoft chaînés entre eux. Il nécessite donc un Apple II+ ou un Apple équipé d'une carte langage ou d'une carte ROM Applesoft. Bien qu'il soit exploitable avec une seule unité de disquettes, il est souhaitable d'en posséder deux afin d'éviter les changements de disquettes rendus nécessaires par les overlays. Le CCA DMS peut être exécuté en 3.2 comme en 3.3

Lors de la première utilisation, le programme demande les données relatives à l'imprimante (slot ou adresse de la routine, données d'initialisation, présence d'un retour chariot -line feed- supplémentaire) et le nombre d'unités de disquettes. Ces données sont stockées dans un fichier CONFIGURATION qui vous évitera de rentrer les mêmes paramètres à chaque exécution du programme. Et il vous suffira de supprimer ce fichier pour repasser automatiquement en phase de configuration.

Menu principal

Il comprend sept fonctions :

- "Terminate processing" : fin (sauvegarde des fichiers et retour au Basic)
- "Perform file maintenance" : entrée de nouvelles données, modification, recherche, suppression...)
- "Print a report or labels" : sorties sur imprimante
- "Compact a file" : compactage du fichier
- "Define a file" : définition du fichier
- "Sort a file" : opérations de tri
- "Change slot number" : changement de slot

Définition du fichier

Chaque champ a trois caractéristiques de base qui doivent être définies :

- un identificateur (1 à 5 caractères) qui sera utilisé dans toutes les opérations pour désigner ce champ;
- un label plus explicite (1 à 15 caractères) qui sera affiché lors de l'entrée des données ou de la sortie sur imprimante;
- la longueur, qui représente le nombre de caractères réservés pour ce champ.

Chaque champ peut avoir trois types différents :

- alphanumérique, c'est-à-dire qui accepte tous les caractères;
- numérique (jusqu'à 9 chiffres) auquel cas la première lettre de l'identificateur doit être une '*';
- permanent, c'est-à-dire fixé dès le départ et non modifiable (intéret. ?!?). La première lettre de l'identificateur est alors un '!'.

Il est également possible de définir des champs calculés à partir des autres champs numériques. La formule entrée est évaluée de gauche à droite, sans hiérarchie entre les opérations. Les parenthèses étant interdites, les possibilités d'exploitation des champs calculés sont assez réduites.

La taille maximale autorisée d'un enregistrement est de 232 octets, et le nombre de champs par enregistrement est limité à 24. La taille totale du fichier ne peut excéder la capacité d'une disquette, ce qui le limite à 140K en DOS 3.3.

Utilisation du fichier

L'option 1 du menu général est utilisée pour ajouter, modifier ou rechercher un enregistrement. La ligne supérieure de l'écran indique à tout moment le nom du fichier, le numéro de l'enregistrement présenté et le nombre total d'enregistrements.

L'entrée des données est effectuée ligne par ligne, certainement par des instructions Basic INPUT, ce qui donne à l'écran une présentation peu esthétique. La fonction Recherche est vraiment minimale, car elle ne permet que la recherche de l'occurence d'une chaîne de caractères dans un champ unique. Il n'y a donc pas la possibilité de tester les valeurs des champs numériques (comparaisons de valeurs) ni d'effectuer des recherches logiques (ET, OU) sur plusieurs critères.

La fonction Modification permet de changer le contenu d'un champ. Cependant, l'enregistrement à modifier doit être spécifié par son numéro, ce qui oblige pratiquement à effectuer une opération de recherche pour connaître le numéro avant chaque modification.

La fonction Suppression présente une caractéristique intéressante : l'enregistrement supprimé est seulement marqué dans le fichier et non pas détruit. Cela permet par exemple de pouvoir imprimer tous les enregistrements supprimés, ou même de les récupérer si le besoin s'en fait sentir ! La suppression effective intervient lors du compactage du fichier (option 3 du menu).

Une dernière fonction permet de créer des fichiers d'index, contenant les numéros des enregistrements satisfaisant à un critère. A partir d'un fichier d'adresses, on peut ainsi constituer un fichier d'index reprenant par exemple toutes les personnes habitant la même ville ou le même département. Ce fichier d'index est utilisé par le "générateur de rapport" qui pourra alors éditer les rensei-

gnements concernant ces personnes.

Edition de rapports

Le "générateur de rapport" permet, à partir du fichier global ou d'un fichier index, d'éditer les données souhaitées. Les paramètres donnés peuvent être les champs à éditer, ou un intervalle de valeurs d'un champ. Il est également possible d'introduire des sous-totaux et des totaux. A noter qu'il est possible d'utiliser une imprimante ayant un driver en RAM, puisque la page 3 est disponible à cet effet ($31A à 3D0).

Possibilités de tris

Il est possible de trier tout le fichier. Bien que nous n'ayons pas pu faire l'essai sur un fichier très gros, il semble que le processus de tri soit particulièrement long : la documentation indique en effet 10 à 15 minutes pour un fichier de 200 enregistrements !

Le tri peut être effectué sur un maximum de 10 champs, et spécifié ascendant ou descendant sur chacun. Si le fichier ainsi trié est supérieur à l'espace mémoire disponible en RAM, le programme utilise un fichier de travail. C'est particulièrement dans ces moments-là que l'on apprécie le fait d'avoir deux drives !

Remarques diverses

CCA DMS et Visicalc étant distribués par la même société, la disquette comprend des programmes de conversion entre CCA et Visicalc, en utilisant le format DIF (Data Interchange Format) qui est devenu un standard pour communiquer avec Visicalc.

CCA DMS étant écrit en Applesoft et non protégé, il est possible de lister tous les programmes et donc de les modifier. De même, les fichiers sont des fichiers TEXT standard et leur structure est expliquée en détail dans la documentation. Il est donc tout à fait possible d'utiliser les données stockées par CCA dans d'autres programmes.

Documentation

La documentation se présente sous la forme d'un classeur comprenant 90 pages de la taille désormais classique des manuels Apple. Elle est rédigée en anglais et permet de découvrir CCA pas à pas grâce à de nombreux exemples.

Conclusion

Le principal avantage de CCA réside dans son "ouverture". Il peut être modifié et ses fichers sont accessibles à tous. Ses principaux défauts sont ses limitations du nombre de caractères par enregistrement (232) et du nombre de champs (24), ainsi que la pauvreté des possiblités de recherche et de tri.

PERSONAL FILING SYSTEM : POUR UNE UTILISATION INDIVIDUELLE

Sur la petite boîte plate qui contient la disquette et la documentation du PFS s'étale un gros macaron bleu sur lequel est inscrit : "New Features". Voici donc la dernière révision du PFS. Qu'y a-t-il de changé ? Le prix tout d'abord, qui a grimpé de 50% à l'occasion du passage à la révision F, mais qui reste tout de même relativement sage pour un programme de gestion de fichier. Les améliorations techniques portent essentiellement sur trois points :

- la possibilité de scinder et fusionner les fichiers;
- celle de redéfinir le fichier en gardant les données déjà entrées;
- l'amélioration des possibilités de tris et d'édition sur imprimante.

Cette dernière amélioration n'est malheureusement pas suffisante pour combler la grande lacune du PFS dans le domaine de l'édition de rapports. Elle ne rend donc pas inutile l'achat du programme "jumeau" : le PFS Report qui a pour unique vocation l'établissement de rapports. Nous avons donc testé l'ensemble PFS et PFS Report afin que le tout soit comparable avec les autres progiciels testés.

Configuration

L'ensemble PFS.Report est proposé sur deux disquettes séparées qui ne peuvent être "bootées" qu'en DOS 3.3. Ces disquettes sont protégées, c'est-à-dire qu'elles ne sont pas copiables. Aucune disquette de sauvegarde n'est incorporée et il faut donc les commander aux USA (15 dollars chaucune).

Le PFS est le seul des quatre progiciels testés ici qui puisse s'exécuter sur un ancien Apple avec mémoire Integer et sans carte langage ou carte ROM Applesoft. L'ensemble du programme est chargé dès le départ dans sa totalité. Il n'y a donc pas d'overlay et il n'est pas nécessaire de garder la disquette programme dans le drive, ce qui autorise l'utilisation du PFS avec un seul drive. Attention cependant, PFS Report nécessite une disquette de travail pour ses tris et rend alors souhaitable le deuxième drive.

Menu principal

Le menu principal comprend six options :

- "Design File" correspond à la définition du fichier;
- "Add" ajoute de nouvelles fiches;
- "Copy" permet la sauvegarde des fichiers,

option rendue nécessaire par le fait que les fichiers créés par le PFS ne sont pas accessibles à partir du Basic;
- "Search/Update" regroupeles fonctions de recherche d'une fiche et de modification;
- "Print" : édition sur imprimante (cette fonction sert très peu lorsque l'on a le Report, beaucoup plus complet);
- "Remove" supprime les fiches.

Le menu principal du PFS Report comprend quant à lui :

- "Print a Report", qui commande l'édition d'un rapport;
- "Pre-define a Report", option permettant de définir à l'avance et de stocker sur disquette la forme et le contenu du rapport;
- "Set New Headings", pour le changement des libellés sur le rapport.

Définition du fichier

Le PFS n'utilise pas le concept d'enregistrements et de champs, mais celui de pages-écrans. Il n'est pas nécessaire de spécifier la longueur et le type de chaque élément. Il suffit de laisser la place adéquate sur l'écran.

La définition s'effectue très facilement en positionnant sur l'écran les différents libellés. On peut ainsi choisir soi-même la présentation de la fiche telle qu'elle apparaîtra à l'écran lorsqu'on voudra la consulter. Le nombre de caractères ou de lignes laissés entre un libellé et le libellé suivant définit la longueur maximum du renseignement qui pourra y être entré (la taille maximale d'un élément est donc légèrement inférieure au nombre de caractères disponibles sur l'écran soit environ 800 caractères).

Le premier élément défini possède une signification particulière puisqu'il sera la clé d'entrée privilégiée dans le fichier. Le temps de recherche de la fiche définie par cette clé est de 2 à 3 secondes. Par contre, toute recherche sur un quelconque autre élément oblige la lecture complète du fichier. Le temps dépend alors de la taille du fichier.

Si vos informations ne tiennent pas sur une page-écran, on peut alors y ajouter des pages additionnelles (jusqu'à 31). La fiche est définie comme étant un ensemble de pages-écrans. Comme d'autre part, il est possible de définir jusqu'à 50 éléments par page-écran, la taille d'une fiche peut être très grande. Seule limitation : le fichier global doit tenir sur une disquette unique, ce qui, dans le cas d'une fiche géante de 32 pages avec 50 éléments par page, ne doit pas représenter beaucoup de fiches ! A l'inverse, si la fiche ne comprend qu'un seul élément, le fichier est quand même limité à 1000 fiches.

Utilisation du fichier

L'option 2 du menu principal permet de remplir les fiches. La présentation de la fiche vierge suit scrupuleusement celle adoptée lors de la définition du fichier. Les libellés sont écrits en inverse, ce qui accroit la lisibilité.

En bas de l'écran sont regroupées deux lignes d'information indiquant le nom du fichier sur lequel on travaille, le numéro de la fiche en mémoire, le numéro de la page-écran affichée et le pourcentage de remplissage de la disquette-fichier (une nouveauté intéressante de la nouvelle révision).

Chaque élément peut être entré séparément, les flèches permettant d'aller rapidement d'un élément à l'autre. Puisqu'il n'y a aucune déclaration des types (alpha, numérique...) des éléments, il n'y a pas de contrôle et il est possible de rentrer n'importe quelle donnée. Le code postal pourra par exemple être rempli avec "ABCDEFGH" !

L'entrée définitive des données a lieu lors de la validation d'une fiche par la touche Control-C. La fiche est alors stockée sur disquette. Il n'y a malheureusement pas de buffer et à chaque entrée d'une fiche correspond un accès disque. Il est possible, avant de valider la fiche, soit d'effectuer des corrections, soit d'annuler purement et simplement l'entrée grâce à la touche ESC. Cette touche ESC est très pratique car elle a la même fonction dans tous les cas : annulation des données ou de la commande déjà frappées et retour au menu.

L'option 4 du menu permet la recherche et la modification d'une fiche. Les fonctions de recherche disponibles sont évoluées, avec notamment la possiblité de comparer le début ou la fin d'un élément, de tester l'occurence d'une chaîne de caractères (avec emploi de "wild caracters" ou caractères nuls), la position d'un caractère... Les tests numériques (inférieur, supérieur, différent, égal) sont également possibles sur les éléments contenant des nombres. Bref, des possibilités qui satisfont à 95% des cas.

Lorsque les critères de recherche sont entrés, le programme affiche la première fiche trouvée. Il est alors possible de la modifier en changeant quelques données, puis de continuer la recherche. Cette facilité d'utilisation a un revers énorme : il n'y a pas de différence entre consultation de fiche et modification, c'est-à-dire que lorsque la fiche est à l'écran, elle peut être modifiée. Il n'existe aucun mode qui permette une

consultation seule, sans risque de la modifier (il suffit d'appuyer, sans s'en rendre compte, sur une touche quelconque pour modifier la fiche !!). Alors attention si d'autres personnes que vous-même ont accès à votre fichier !

Edition de rapports

L'option 5 du PFS permet l'édition des fiches sur imprimante. Cette édition peut se faire sur une partie du fichier grâce à la spécification des fiches qui doivent être prises en compte, spécifications effectuées à l'aide des fonctions de recherche définies plus haut. Ensuite sont définis les éléments à éditer. Le PFS a des possibilités de définition et de spécification de rapports très réduites; le PFS Report est heureusement là pour le seconder.

L'option 3 du Report permet de remplacer sur les rapports les libellés définis dans la fiche par des libellés plus explicites.

L'option 2 concerne la prédéfinition des rapports. Ces définitions sont stockées sur la disquette et on peut mémoriser 8 types de rapports simultanément. Elles indiquent les éléments qui doivent être édités, l'ordre dans lequel ils seront imprimés et la mise en page du rapport (sauts de pages, sauts de lignes...). Le tri a lieu sur les deux éléments les plus à gauche sur le rapport. Il peut être spécifié en alphanumérique (le tri est réalisé selon l'ordre alphabétique) ou numérique (le tri est fait par ordre décroissant). Il n'est notamment pas possible d'effectuer un tri numérique croissant ou un tri dans l'ordre inverse de l'alphabet.

Le PFS Report autorise l'introduction de lignes supplémentaires consistant en des calculs de totaux, de nombres d'éléments dans une colonne ou de moyennes. Ces calculs peuvent être effectués sur le rapport tout entier, ou sur chaque page du rapport sous forme de sous-totaux. Le Report autorise également la définition de colonnes dérivées (3 au maximum) dans lesquelles pourront être reportés des calculs numériques effectués à partir des autres colonnes. Ces colonnes dérivées ne doivent pas être confondues avec des champs calculés, car elles n'existent que sur le rapport et il n'est pas possible de les incorporer dans la fiche.

Divers

La nouvelle révision ajoute au PFS un certain nombre de fonctions supplémentaires dont deux sont les bienvenues : la redéfinition du fichier et la possibilité de scinder ou fusionner les fichiers. La redéfinition du fichier permet d'adapter le fichier aux besoins au fur et à mesure du temps. Certains éléments définis sont peut-être inutiles, d'autres font cruellement défaut. Lors d'une redéfinition, les données correspondant aux libellés du nouvel écran présents dans l'ancien, seront reprises dans le nouveau fichier.

Quant à la séparation d'un fichier en plusieurs morceaux ou la fusion de plusieurs fichiers, elle permet une meilleure adaptation du fichier aux besoins réels.

Documentation

Le PFS et le PFS Report disposent chacun de leur manuel (90 et 45 pages respectivement). Ecrits en anglais, très clairs, ils comprennent de nombreux exemples et surtout beaucoup de représentations de l'écran. Mais le plus agréable, c'est la version française du manuel de l'opérateur jointe avec le PFS. Cette brochure de 45 pages, éditée par Sanotec (l'importateur de PFS), a été visiblement tirée en traitement de texte sur Apple et une Centronics. Le résultat est bon et réjouira tous ceux à qui l'anglais donne des maux de tête.

Conclusion

La qualité primordiale du PFS, c'est avant tout sa facilité d'utilisation (surtout pour la partie définition du fichier) et sa rapidité d'apprentissage. Dans le cas d'une utilisation personnelle sans gros besoins de rapports, il n'est peut-être pas nécessaire d'acheter le PFS Report; le prix est dans ce cas très compétitif par rapport à celui des autres programmes.

S'il est indiqué pour des utilisateurs individuels, il est par contre peu recommandé pour des applications professionnelles en raison principalement de l'absence de contrôle lors de la saisie et surtout de l'impossibilité de consultation sans risque de modification.

VISIFILE : POUR VISI-PHILES UNIQUEMENT ...

Nous voici en pleine Visi-manie. Après Visicalc, Visidex, Visitrend, Visiplot, Visimachin, voici venir Visifile distribué comme il se doit par Visicorp, nouvelle dénomination de Personal Software !

Ouvrons donc le classeur. A la fin, dans une pochette en plastique, deux disquettes. Un original et un backup ? Excellente idée ! Mais non, vous n'y êtes pas : il n'y a pas de backup et le programme est réparti sur deux disquettes. Comme trop souvent, vous devrez écrire aux USA et trouver quelques dollars pour pouvoir bénéficier de la disquette de backup.

Dans cette même pochette est incorporée une carte de référence à cinq volets. Voilà qui va nous être bien utile tout au long de cet essai !

Configuration

Il est possible d'exécuter Visifile sur n'importe quel type d'Apple II (Applesoft ou non), pourvu qu'il soit équipé du DOS 3.3. Au premier chargement du programme, il est nécessaire d'inscrire le nombre de drives connectés. Le numéro du slot est testé directement par le programme (quel raffinement !) qui constitue le classique fichier "configuration" évitant de rentrer les paramètres à chaque exécution du programme. Les overlays étant nombreux, il parait souhaitable de disposer de deux drives : un pour le programme et un pour les fichiers.

Menu principal

Après affichage de la version et du numéro de série, le menu principal est affiché à l'écran. Il n'occupe que quatre lignes en haut de l'écran, afin de laisser le reste de l'écran libre pour les fiches ou pour l'affichage des commandes disponibles qui se trouvent ainsi pratiquement toujours sous les yeux de l'utilisateur. Les deux flèches permettent de sélectionner la commande désiréee à l'aide d'un curseur en vidéo inverse (il me semble avoir déjà vu cela quelque part...)

On dispose de neuf options :

- "Select" : choix du fichier
- "Define" : définition du fichier
- "Maintain" : entrée, consultation, modification et suppression des enregistrements.
- "Sort" : création des fichiers d'index, afin d'ordonner le fichier principal
- "Copy" : duplication d'une partie ou de la totalité de la disquette fichier
- "Print" : édition sur imprimante
- "Back up" : réalisation de disquettes de sauvegarde pour les fichiers (pratiquement identique à l'option Copy).
- "Special" : fonctions annexes telles que l'initialisation de disquettes vierges, le transfert de fichiers DIF, le changement des paramètres disques et imprimante...
- "Done" : fin de la séance de travail.

Définition du fichier

Elle est réalisée par l'intermédiaire de la fonction "Define". Lorsque cette fonction est choisie au moyen du curseur et, après quelques tours de disquette sur chaque drive, un sous-menu est affiché. La fonction "Create" permet de créer et de définir le fichier. Pour chaque enregistrement doivent être définies trois caractéristiques :

- le nom, limité à 10 caractères
- la longueur, limitée à 128 caractères pour un champ alphanumérique et à 38 pour un champ numérique
- le type : Alphanumérique, Numérique ou Permanent (?!?)

Comme pour le CCA DMS, la taille maximum d'un enregistrement est de 232 caractères (bien faible!) ou de 24 champs. Le nombre de caractères disponibles est affiché en permanence en haut de l'écran et est diminué au fur et à mesure de la création des champs.

Une caractéristique intéressante est à noter : il est possible d'opter lors de la définition du fichier pour un format de visualisation automatique ou personnalisé. Dans l'option automatique (par défaut), chaque champ est placé aussi près que possible de son prédecesseur. Si la place restant sur la ligne n'est pas assez importante, il est placé sur la ligne suivante. Dans le cas du format personnalisé, l'utilisateur doit spécifier la place qu'il souhaite attribuer sur l'écran à chacun des champs.

Utilisation du fichier

Une fois l'option "Maintain" choisie, l'utilisateur peut alors assister à un Son et Lumière spécial pour deux drives. L'écran dispense des messages rassurants, soit dans l'ordre : "Reading...","Writing..."(tiens, tiens, quelle idée ?!),"Loading Maintenance functions" (là, j'ai de la chance car toutes les fonctions se trouvent sur la même disquette programme),"Reading files, please wait"(je ne fais que cela!), ,"Formatting screen"... En tout 28 secondes bien longues !

Une fiche vierge est alors présentée à l'écran. On peut à présent ajouter une fiche, la modifier, la supprimer ou chercher une autre fiche, tout ceci heureusement sans trop de temps d'attente. L'entrée est relativement facile grâce aux deux flèches qui permettent de passer d'un champ à un autre sur l'écran. Sont également utilisables de nombreuses fonctions d'édition (sous forme de caractères de contrôle) qui permettent par exemple d'insérer ou de supprimer un caractère.

Les fiches ne sont pas enregistrées sur la disquette au fur et à mesure de leur entrée, mais à une fréquence dépendant de leur longueur. A noter que pour la modification ou la recherche d'un fiche, il est nécessaire de fournir son numéro, ce qui n'est pas très commode.

Possibilités de tri

Le tri d'un fichier Visifile provoque la création d'un fichier d'index qui pourra ensuite servir à l'édition de rapports ou à la consultation de fiches. Il est possible de trier jusqu'à 10 champs maximum et de stocker un maximum de 24 fichiers d'index pour chaque fichier Visifile.

Divers

Etant édité par Visicorp, Visifile dispose évidemment d'une structure de fichier au

format DIF qui lui permet d'utiliser les données d'autres programmes et notamment de récupérer les fichiers de CCA DMS.

Documentation

Ecrite en anglais, la documentation de Visifile est l'un de ses points forts. Grâce à un exemple de fichier fourni sur la disquette programme, l'utilisateur novice peut rapidement apprendre le fonctionnement de Visifile en suivant les différents chapitres de la documentation, dans laquelle on trouve de très nombreuses représentations de l'écran.

Conclusion

La dernière pétale de la marguerite que j'effeuillais est tombée sur "pas du tout". C'est vrai, j'ai été quelque peu déçu par Visifile. Certes, on ne peut pas refaire tous les jours un programme de la trempe de Visicalc. Mais j'ai retrouvé dans Visifile beaucoup trop de limitations du CCA DMS telles que la longueur des champs et de l'enregistrement ou la faiblesse de l'évaluation des formules pour les champs calculés évoquée pour CCA. Et ce que l'on pardonne à un programme vieux de trois ans ne peut plus l'être pour un programme récent. Quant à la lenteur des opérations, elle a mis mes nerfs à bout. Attendez donc la version suivante et pensez à PFS ou DB Master pour vos achats !

DB MASTER : LE MAITRE ...

Un nom qui "sonne" bien, et annonce d'entrée la couleur : Master Stoneware, la société qui le distribue, ne fait pas dans la nuance. Mais il est vrai que l'on peut se permettre d'être ambitieux avec un tel produit !

Comment vous le présenter ? A vrai dire, l'Ordinateur Individuel, revue bien connue, m'a coupé l'herbe sous les pieds puisqu'ils ont publié un essai du DB Master dans le numéro du mois de juillet-août 1982. Je ne serai pas rancunier et je vous recommande de lire leur article avant le mien afin de faire connaissance avec le DB. Il n'est donc pas question de refaire ici une analyse complète, mais plutot de souligner quelques points et d'examiner les deux disquettes utilitaires prévues pour le DB.

Atouts et points faibles du DB Master

L'une des premières choses qui frappe, lorsqu'on débute sur le DB, c'est l'utilisation de mots de passe. Non pas un. mais trois qui permettent une hiérarchie entre les différents utilisateurs et assurent une certaine sécurité et confidentialité des données. Le premier mot de passe n'autorise que la lecture des champs non protégés à la conception, le second permet la lecture et la modification de tous les champs existant dans un enregistrement. Le troisième est le sésame : il permet de changer tous les mots de passe et donc de sélectionner les utilisateurs en leur fournissant le nouveau mot de passe ou non. Rassurez-vous, si vous avez la mémoire défaillante, vous pouvez toujours concevoir votre fichier sans aucun mot de passe !

Autre surprise, la mention "illimitée" en face de l'information taille maximum du fichier. En effet, DB peut stocker les données d'un fichier sur autant de disquettes qu'il le faut. Cette possibilité de chaînage des disquettes, couplée à la compression importante des données avant leur stockage, autorise la création et l'exploitation de fichiers importants, dont la taille ne sera plus limitée que par l'ennui de la personne chargée de changer les innombrables disquettes dans le drive (bon courage : la documentation signale que l'on peut aller jusqu'à 256 disquettes !).

Chaque enregistrement est défini par une clé primaire qui doit être unique (35 caractères maximum) et peut posséder plusieurs clés secondaires. Grâce à ces clés secondaires, on peut retrouver rapidement une fiche à partir de renseigenements divers. En effet, le temps maximum de recherche d'un enregistrement par sa clé primaire est de trois secondes, celui de la recherche à partir d'une clé secondaire est de 5 à 7 secondes. Mais cette performance se paie cher : le moindre fichier nécessite deux disquettes ! La première contient le fichier de chaînage des clés secondaires par rapport aux clés primaires ainsi que les rapports pré-définis. La seconde contient les données proprement dites. Comme, de plus, la disquette programme doit rester en place en raison des overlays, le minimum confortable pour l'utilisation de DB est de 3 drives (et donc deux controleurs).

Le DB autorise 10 types de champs différents. La définition d'un fichier, qui est déjà compliquée avec la spécification des clés primaires et secondaires, s'en trouve encore alourdie. Mais ce n'est qu'une mauvaise étape à passer car, une fois le fichier défini, l'utilisation est beaucoup plus facile et plus sûre. Il n'est plus possible de se tromper dans les entrées, car tous les contrôles sont faits : type des données, présence facultative ou obligatoire, longueur fixe ou variable... Il est même possible d'utiliser des valeurs par défaut afin de faciliter la tâche. Parmi les types de champs les plus intéressants, on dispose de :

- "L'Auto-date", qui contient la date à laquelle l'enregistrement a été accédé pour la dernière fois. Cette date est remise à jour à chaque accès de l'enregistrement. On peut toutefois déplorer, comme pour le type

→

	CCA DMS	PFS	PFS REPORT
Nom du programme...	CCA Data Management System	Personal Filing System	Personal Filing System Report
Auteur(s).....	Helmar Ben Herman & Colin G. Jameson	John Page & D.D Roberts	John Page
Distribution.....	Personal Software	Software Publishing Corporation	Software Publishing Corporation
Version testée.....	CCA DMS V 5.2	PFS Rev F (03/82)	Rev C (08/81)
Prix indicatif.....	1000 F	Rév E : 880 F Rév F : 1250 F	880 F
Documentation.....	Anglaise, 90 p.	Anglaise, 90 p. Française, 45 p.	Anglaise, 45 p.
Disquette..... Backup..... Achat Backup.....	Non protégée - -	Protégée Sans Backup USA : 15 $	Protégée Sans Backup USA : 15 $
Configuration Apple	Apple II + ou Apple II avec ROM Applesoft ou carte langage	Apple II ou Apple II+	Apple II ou Apple II+
Nombre de drives souhaitables.....	2	1	2
Fichier maximum....	1 disquette	1 disquette	-
Taille maximum des enregistrements...	232 caractères ou 24 champs	32 pages-écrans	-
Taille maximum d'un champ.....	232 caractères	1 page-écran soit 800 caractères	-
Type des champs....	Alpha, Numérique et Permanent	pas de définition de types	Alpha et Numériques
Champs calculés....	Oui	Non	3 : uniquement sur le rapport
Compatibilité DIF..	Oui	Non	-
Possibilité de Redéfinition..... Fusion fichiers..	 Non Non	 Oui Oui	 - -

	VISIFILE	DB MASTER	UTILITY PAK 1&2
Nom du programme...	Visifile	Data Base Master	-
Auteur(s).....	Richard Ewing & John Unger Zussman	Barney Stone	Barney Stone
Distribution.....	Visicorp	Stoneware	Stoneware
Version testée.....	Visifile V 1.0	Version 3	-
Prix indicatif.....	2395 F	2300 F	UP #1 : 995 F UP #2 : 1200 F
Documentation.....	Anglaise, 164 p.	Anglaise, 130 p.	UP #1 : 48 p. UP #2 : 18 p.
Disquette..... Backup..... Achat Backup.....	Protégée (2 Dk) Sans Backup USA : 20 $	Protégée Sans Backup USA : Gratuit	Protégée Backup Fourni -
Configuration Apple	Apple II + ou Apple II	Apple II + ou Apple II avec ROM Applesoft ou carte langage	Apple II + ou Apple II avec ROM Applesoft ou carte langage
Nombre de drives souhaitables.....	2	3	3
Fichier maximum....	1 disquette	illimité	-
Taille maximum des enregistrements...	232 caractères ou 24 champs	1020 caractères ou 100 champs	-
Taille maximum d'un champ.....	Alpha : 128 car. Num : 38 car.	Alpha : 30 car. Num : 9 car.	-
Type des champs....	Alpha, Numérique et Permanent	Alphanumérique Numérique (3) Date (2) Dollar/cents Y/N - Numéro Sécurité Sociale	-
Champs calculés....	Oui	24 : uniquement sur le rapport	-
Compatibilité DIF..	Oui	Oui	-
Possibilité de Redéfinition..... Fusion Fichiers..	 Oui Non	 Oui avec UP #1 Oui avec UP #1	 - -

"Date", qu'il n'y ait pas le choix entre la notation européenne (jour/mois/an) et la notation américaine (mois/jour/an).

- Le type Y/N, ne pouvant prendre que deux valeurs : Yes ou No.

- Le type "Dollar/Cents" tout-à-fait adapté aux données exprimées en dollars. Seul problème, en francs et centimes bien de chez nous, la limitation à 9 digits correspond à une valeur sept fois plus faible qu'en petits billets verts !

Les possibilités de tri paraissent sans failles puisqu'il est même possible d'effectuer des ET et des OU logiques entre les différents critères choisis. Mais là encore, le nombre de possibilités et la densité de l'affichage à l'écran rendent la maîtrise du système longue à acquérir. Et l'absence d'une carte récapitulative de toutes les commandes se fait largement sentir.

Autre invention bien pratique : les "short forms" destinées à faciliter la consultation du fichier ou l'entrée des données. Le principe est de ne présenter à l'écran que les champs utiles à l'utilisateur, sous une forme qu'il aura lui-même définie, et d'éliminer toutes les autres qui sont inutiles. Cette possibilité accroit notablement la lisibilité du fichier et permet surtout une utilisation plus facile pour des personnes autres que le concepteur.

Stoneware a d'autre part réalisé deux disquettes d'utilitaires qui complètent les fonctions du DB. La première date de 1981, tandis que la seconde vient juste de sortir. Dans les deux cas, ces disquettes sont protégées et donc non copiables, mais une disquette de sauvegarde (backup) est présente avec l'original (c'est trop rare pour ne pas le signaler !).

La disquette Utility PAK 1

Elle ajoute un nombre important de possibilités au DB :

-redéfinition et restructuration du fichier avec la possibilité d'ajouter ou d'enlever des champs, de modifier les champs calculés et surtout de redéfinir la clé primaire. Toutes ces fonctions sont d'autant plus importantes sur le DB que les contrôles sont très stricts en ce qui concerne les types des champs et que la moindre erreur de définition du fichier peut avoir à la longue des conséquences néfastes.
-duplication des fichiers de données afin de constituer des sauvegardes.
-fusion totale ou sélective de 2 fichiers.
-possibilité de "réparer" une disquette de données endommagée (inutile si on a fait consciencieusement les sauvegardes nécessaires...).
-édition de statistiques sur les fichiers (déjà incorporé dans la disquette programme du DB Master.
-translation des fichiers afin de les rendre compatibles avec le format DIF (utilisé par Visicalc). A cet effet, la documentation inclut la spécification technique complète du format DIF. Autre possibilité de translation possible : fichier DB vers fichier TEXT et inversement. Une idée intéressante si on veut utiliser les données du DB pour d'autres programmes.

La disquette Utility PAK 2

Au contraire de la disquette 1 qui n'est valable que pour la version "standard" du DB, la disquette 2 peut également être utilisée avec les versions spéciales pour disques 8 pouces et disques durs connectables à l'Apple (notamment pour le Corvus 5 et 10 Mo). Les fonctions offertes sont les suivantes :

- "Global Editor" qui permet de supprimer ou de modifier le contenu de certains champs sur le fichier entier. Il est par exemple possible de remplacer le contenu d'un champ par une constante ou le résultat d'un calcul, d'incorporer dans un champ numérique la différence entre deux dates présentes dans l'enregistrement...
- "The Label Printer" améliore et facilite l'édition d'étiquettes. Il autorise la conversion automatique des données issues du fichier DB en minuscules, sauf pour la première lettre du mot qui reste en majuscule.
- "Transaction File Merge" apporte la possibilité de transférer le contenu d'un ou plusieurs champs numériques entre 2 fichiers.
- "Reblock for Extra Space" est particulièrement utile lorsque la taille du fichier dépasse une disquette. Il est alors intéressant de regrouper toutes les informations relatives à un même enregistrement sur une disquette unique afin d'éviter d'avoir à changer les disquettes trop souvent.
- "Change Read Protected Fields" permet de changer les champs protégés en champs normaux et inversement.

Conclusion

Le DB Master est à mes yeux le meilleur programme disponible sur Apple II (sans la carte Z80) pour une application professionnelle. Doté de ses deux disquettes d'utilitaires, il représente un fantastique outil de travail. Mais attention, la puissance se paie par une complexité plus grande et il faudra un certains temps d'adaptation pour assimiler son maniement. Le surnom américain de DB Master est, ne l'oublions pas, "DB Monster"! Le seul reproche qui peut être fait quant à ses caractéristiques techniques concerne la longueur maximale d'un champ alphanumérique,de 30 caractères, qui peut se révéler dans certains cas insuffisante.

DES LIVRES POUR VOTRE APPLE II

MATÉRIELS

La découverte de l'Applesoft

Tome 1
par Dominique Schraen et Frédéric Lévy

Cet ouvrage d'initiation s'adresse aussi bien aux futurs utilisateurs de l'Apple voulant apprendre la programmation en Basic Applesoft, qu'à l'Applelophile chevronné sollicité par ses proches curieux de "voir un peu comment ça marche". D'approche progressive, il est illustré de nombreux exemples et exercices.

Série verte
128 pages - 65,00 FF/500,00 FB

La découverte de l'Applesoft

Tome 2
par Frédéric Lévy

Recueil d'exercices destiné à tous ceux qui connaissent les instructions Basic de l'Applesoft et ne maîtrisent pas encore la programmation, c'est une invitation à l'analyse et à la programmation de problèmes simples et fréquemment rencontrés. L'énoncé de chaque exercice est suivi de son analyse : une ou deux solutions commentées sont proposées.

Série verte
120 pages - 65,00 FF/500,00 FB

La pratique de l'Apple II

Volume 1
par Nicole Breaud-Pouliquen

Cet ouvrage présente les spécificités du Basic Applesoft à partir d'une description du matériel et du logiciel du système Apple. Les techniques de programmation, de composition et d'animation de dessins et graphiques colorés y sont expliqués à l'aide d'exemples illustratifs et d'exercices résolus.

Série bleue
128 pages - 65,00 FF/500,00 FB

La pratique de l'Apple II

Volume 2
par Nicole Breaud-Pouliquen

Ce second volume de la pratique de l'Apple II est consacré au système d'exploitation disque, à la gestion des fichiers, à l'impression et aux imprimantes, à la carte horloge Applecloc. De nombreux exemples de programmes illustrent les fonctions et les commandes décrites.

Série rouge
120 pages - 65,00 FF/500,00 FB

La pratique de l'Apple II

Volume 3
par Nicole Breaud-Pouliquen et Daniel-Jean David

Ce volume est une initiation à la programmation en langage machine 6502, dont le jeu d'instruction est expliqué et utilisé. L'assemblieur symbolique et ses logiciels connexes y sont décrits. L'interaction avec le Basic et avec le système y sont étudiés.

Série noire
176 pages - 75,00 FF/570,00 FB

GUIDES PRATIQUES

LISP sur Apple II

par Nicole Breaud-Pouliquen

Description concrète et progressive de la programmation en langage LISP sur l'ordinateur Apple II, ce livre démystifie et met en évidence la puissance à l'expression de ce langage.
De nombreux exercices et la présentation d'exemples complexes appliqués à la gestion des listes, l'analyse grammaticale et l'élaboration de dessins récursifs complètent cet exposé.

Série bleue
112 pages - 65,00 FF/500,00 FB

PROGRAMMES

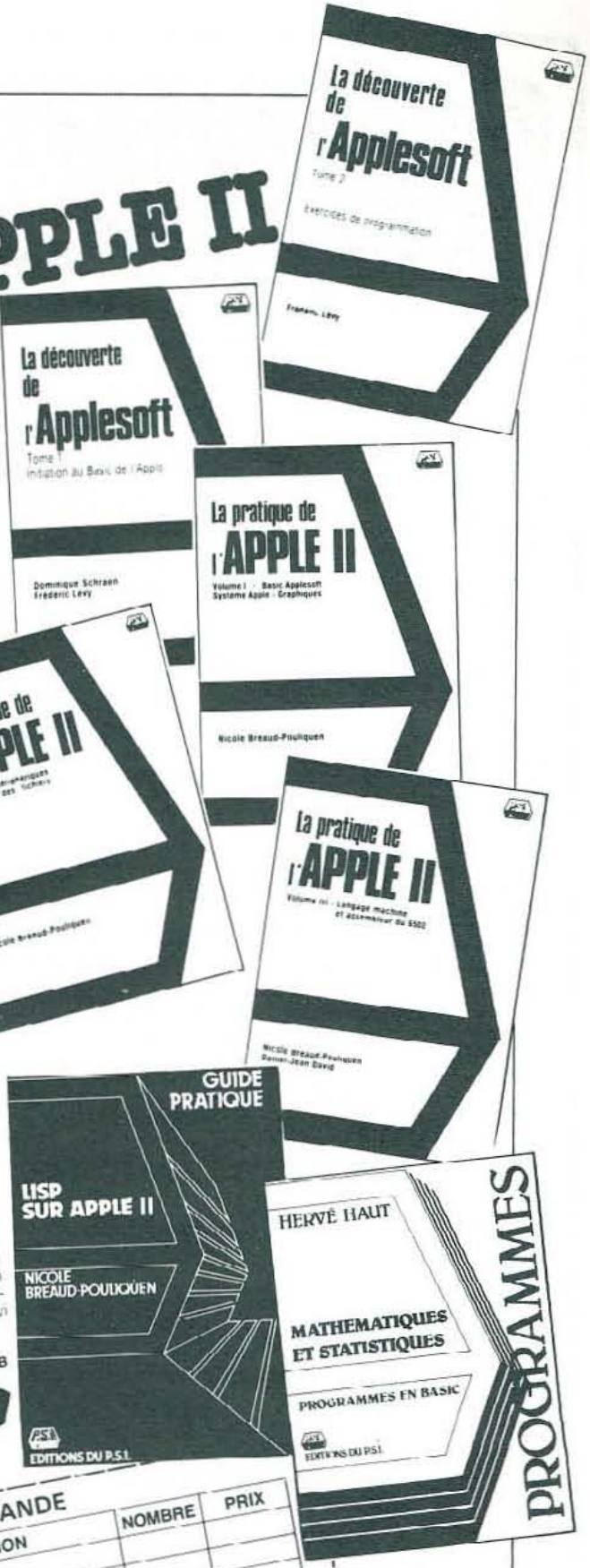
Mathématiques et statistiques

Par Hervé Haut

Cet ouvrage est un recueil de 16 logiciels de base (niveau supérieur) tant en mathématiques qu'en statistiques. Chaque problème traité comporte une introduction numérique, un exposé de la technique de programmation utilisée, un organigramme détaillé et un programme complet en Basic suivi d'un exemple d'utilisation.

Série rouge
272 pages - 85,00 FF/650,00 FB

NOUVEAU



P.S.I. DIFFUSION

P.S.I. DIFFUSION
41-51, rue Jacquard
BP 86 - 77400 Lagny-s/Marne
FRANCE
Téléphone (6) 007.59.31

P.S.I. BENELUX
5, avenue de la Ferme Rose
1180 Bruxelles
BELGIQUE
Téléphone (2) 345.08.50

BON DE COMMANDE

DESIGNATION	NOMBRE	PRIX
	TOTAL	

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg, à P.S.I. BENELUX

Les prix sont : taxes, emballage et port compris.
(par avion : ajouter 8 FF (75 FB) par livre)

A.PO

NOM _____ PRENOM _____ N° _____

rue _____ Ville _____

Code postal _____

Date et signature

ACGRAPH

La programmation facilitée

Denis Sureau

Le langage d'assemblage permet d'écrire des programmes très rapides mais aussi - le moniteur est écrit dans ce langage - de créer les fonctions d'un ordinateur, aussi la possession d'un assembleur (LISA, BIG MAC, SC) peut nous inciter à jouer les Steve Wozniak et doter notre Apple de fonctions nouvelles. Dans un programme Applesoft, certaines procédures (formatage de l'écran, menu, boucle d'attente) reviennent constamment, et notre imagination créatrice se tournera plutôt vers ces fonctions...

L'instruction INPUT de l'Applesoft, refuse obstinément "," et ":" que ce soit à partir du clavier ou du disque. Nous détournerons avant tout cet inconvénient en utilisant la procédure GETLIN(FD6A) du moniteur, tandis que PTRGET(DFE3) cherchera le nom de la variable spécifiée dans la table et, si elle ne la trouve pas, l'y ajoutera.

Pour transférer le contenu du buffer d'entrée ($200) vers la mémoire, il faut trouver un emplacement libre parmi les autres variables, avant d'effectuer le transfert; mais si l'adresse du nom de la variable dans la table se trouve en FORPNT(85,86) déjà obtenu grâce à PTRGET(DFE3) précédemment employé, et si l'on charge les registres A et Y avec l'adresse du buffer d'entrée et X avec le caractère RETURN($8D) qui délimite la fin de la chaîne, alors la procédure MAK (E3E9) de l'Applesoft fera tout cela pour nous! SAVD (DA9A) placera l'adresse de la chaîne en mémoire et sa longueur dans la table des variables.

Ainsi est née l'instruction & INPUT , et elle ne refuse pas les virgules! Cependant, l'emploi de fichiers à accès direct par exemple demande que les données aient une longueur limitée et, pour éviter les problèmes causés par des données amputées de la partie excédentaire, une boucle affichant une ligne de points de la longueur désirée, grâce à la procédure d'affichage COUT ($FDED) , nous dotera d'une authentique instruction de formatage; mais auparavant une autre boucle contenant cette fois BS (CALL-1008) (FC10), qui recule le curseur, placera celui-ci en début de ligne.

Notre instruction prendra maintenant la forme: & INPUT A ,Ll où l représente (à titre indicatif) la longueur maximale des données à entrer.

On peut aussi éviter l'emploi des instructions VTAB et HTAB en définissant deux paramètres supplémentaires et en plaçant la position verticale (-1) à l'adresse $25 (37), en appelant la procédure VTAB du moniteur, et en spécifiant la position horizontale à l'adresse $24 (en décimal 36 et toujours -1, la soustraction étant faite par le programme assembleur).

Sous sa forme la plus complète l'instruction devient: &INPUT A ,Ll,Vv,Hh,"LABEL" où l est la longueur de l'information, v représente la ligne d'écran (comprise entre 1 et 24), h la colonne (de 1 à 40) et LABEL, toujours en dernière position, un texte à afficher avant interrogation du clavier. L'ordre des trois paramètres numériques est libre, mais il est nécessaire que la variable, une variable de chaîne de caractères, simple ou dimensionnée, soit analysée en premier lieu. Voila une instruction de formatage d'écran que l'on pourrait appeler par exemple FORMAT, qui facilite l'entrée de données; nous verrons plus loin qu'avec quelques instructions Applesoft on obtient aisément l'équivalent en sortie, nécessitant normalement un PRINT USING.

Attaquons-nous pour l'instant à la deuxième fonction suggérée en introduction, la sélection d'une option dans un menu. Il existe un système de sélection qui évite de numéroter les différentes options et donne à un menu un aspect parfaitement professionnel. Un curseur se déplace d'une option à l'autre; il peut s'agir d'une flèche qui se positionnera devant chaque item ou d'une barre inversée qui présentera en noir sur blanc l'option sur laquelle on la positionne. A cet instant, on appuie sur RETURN et l'option spécifiée de la sorte est sélectionnée, grâce en l'occurence à la procédure en assembleur que nous allons décrire.

Pour réaliser la barre, nous ferons appel à VTAB(FC22) qui calcule l'adresse d'une ligne d'écran à partir d'un numéro placé en CV($25) et place une adresse en $28 et $29. La marge gauche et la longueur de la barre étant spécifiées, on chargera chaque position d'écran en A, l'instruction AND #$3F convertissant le code du caractère en son équivalent inversé qui sera stocké sur la même position.

Lorsqu'on appuie sur la flèche droite du clavier, le curseur doit se positionner sur l'item suivant et, lorsqu'on appuie sur la flèche gauche, remonter au précédent. Comme le nombre d'interlignes peut varier d'un menu à l'autre, on spécifiera par POKE le nombre de sauts de lignes, normalement deux à l'origine, la table des PATCHES précisant plus loin à quelle adresse. Elle indique également comment remplacer la flèche, symbolisée par le caractère ">", par un autre caractère quelconque.

Notre objectif étant de simplifier le menu au maximum une instruction particulièrement riche viendra se placer entre la liste des options et le branchement vers les sous-programmes correspondants, et rien d'autre! Son format est le suivant:

& LET A, VTAB , MARGE , LONGUEUR , NOMBRE D'ITEMS

Ici, l'ordre des paramètres est fixe, et ce sont respectivement le nom de la variable qui contiendra le numéro de l'option sélectionnée, le numéro de ligne de la première option, la marge, la longueur de la barre, qui sera remplacée par une flèche si l'on spécifie 0 et le nombre d'options dans le menu.

Lors de la sélection, si l'on tape ESCAPE, l'instruction HOME s'exécutera et le programme s'arrêtera. Une sélection numérique basée sur l'ordre relatif des items sera également possible, afin de prévoir tous les cas. D'ailleurs, l'usage de cette routine ne se limite pas aux menus, et je ne doute pas que vous en trouviez d'autres utilisations!

S'ils sont placés dans l'ordre, avec leurs numéros de départ espacés par un incrément égal, plusieurs sous-programmes seront appelés en Applesoft par une instruction du genre ON X GOTO 1000, 2000, 3000, etc..., en Integer par l'équivalent GOTO X * 1000, plus simple, comme il est également plus facile de se souvenir d'un nom de sous-programme que d'un numéro de ligne. Les instructions & GOTO ou & GOSUB que l'on rajoute à notre programme nous donnent cette possibilité de l'Integer.

La dernière fonction que nous envisagions de programmer était la boucle d'attente et nous utiliserons pour cela la procédure WAIT ($FCA8) du moniteur. Ce que nous voulons, c'est une instruction simple et & WAIT devrait faire l'affaire! Pour les impatients, on prévoira la possibilité de poursuivre le traitement dès que qu'une touche est enfoncée. Placée à l'intérieur d'une double boucle, avec le test du clavier, la procédure $FCA8 engendre une attente de 15 secondes que l'on peut modifier en agissant par POKE sur le nombre de tours d'une des boucles, comme le montre le tableau des PATCHES ci-dessous.

TABLEAU DES PATCHES

POKE 37656,X	Saut de X lignes. X=1 si l'on veut un menu sans interligne
POKE 37657,X	Flèche : X est le code ASCII du caractère à afficher + 128
POKE 37658,X	Durée d'attente : X=128 augmente la durée, X=128 diminue

(pour 32K, soustraire 16384 à ces valeurs)

Notre programme assembleur ne contient hélas pas de fonction PRINT USING, bien utile pour formater l'écran en sortie, mais les opérations de base de cette instruction peuvent s'obtenir simplement en Applesoft:

HTAB 30 - LEN(STR (INT(A)));

permet d'aligner une colonne de nombres sur la virgule ou, si ce sont des entiers, de les aligner sur le chiffre des unités tandis que l'arrondi au plus proche s'obtient en définissant une fonction ARR:

DEF FN ARR (A)= INT(A*100+.5)/100

qui arrondit ici sur deux chiffres, et en affichant: PRINT ARR(A)

L'arrondi au chiffre inférieur s'obtient en supprimant +.5 dans la définition ci-dessus.

Le programme en BASIC listé ci-dessous est un programme de démonstration appelant la version 48K du programme. Le code source et la récapitulation en hexa de la version 48K du programme, ainsi que la récapitulation de la version 32K, sont fournis dans la suite de cet article.

```
]LIST

1  REM  ************************
        * -------------------- *
        *    PROGRAMMATION     *
        *      FACILITEE       *
        *        --            *
        *    PAR D. BUREAU     *
        *      (C) 1982        *
        * -------------------- *
2  REM  ************************
3  REM
4  PRINT  CHR$ (4)"BRUN PROGR48K"

5 APP$ = "AMPER WAIT...."
6 INPT = 10000
15  TEXT : HOME
16  FOR I = 1 TO 40: PRINT "&";: NEXT
     I
17  PRINT "&";: HTAB 40: PRINT "&
     &";: HTAB 40: PRINT "&&";: HTAB
     40: PRINT "&";
18  FOR I = 1 TO 40: PRINT "&";: NEXT
     I
```

```
19  REM  ----------------------
          PRESENTATION MENU
        ----------------------
20  PRINT : VTAB 3: HTAB 11: PRINT
     "DEMO MENU RAPIDE"
23  VTAB 6
25  PRINT "----COMMANDES---<";: INVERSE
     : PRINT " ESC <   > RTN 1-9 "
     ;: NORMAL
26  PRINT ">--"
27  PRINT : PRINT "   VOTRE CHOIX?
     "
30 VTB = 11: VTAB VTB
40  PRINT "       ITEM 1"
45  PRINT
50  PRINT "       ITEM 2"
55  PRINT
60  PRINT "       ITEM 3"
65  PRINT
70  PRINT "       ITEM 4"
100  &  LET A,VTB,5,33,4
108  REM
109  REM  DANS L'ORDRE...
110  REM  A=NUMERO CHOISI
111  REM  1-VTAB DU 1ER TITRE
112  REM  2-MARGE GAUCHE
113  REM  3-LARGEUR DU CURSEUR
          INVERSE
115  REM  4-NOMBRE DE TITRES
116  REM L'ORDRE DES PARAMETRES
117  REM  EST FIXE.
490  REM  ---------------------
          S E L E C T I O N
        ----------------------
500  &  GOSUB A * 1000
550  PRINT
560  PRINT APF$: &  WAIT
600  &  GOTO INPT
900  REM  ---------------------
          E X E C U T I O N
        ----------------------
990  REM SOUS-PROGRAMMES A
991  REM EXECUTER...
1000  GOSUB 5000: PRINT "ITEM 1": RETURN
2000  GOSUB 5000: PRINT "ITEM 2": RETURN
3000  GOSUB 5000: PRINT "ITEM 3": RETURN
4000  GOSUB 5000: PRINT "ITEM 4": RETURN
5000  VTAB 20
5010  PRINT "...EXECUTION DE L'";:
      RETURN
9990  REM ---------------------
          F O R M A T
        ----------------------
10000  TEXT : HOME
10010  PRINT  TAB( 11)"** F O R M
     A T **"
10030  PRINT : PRINT
10040  PRINT "INPUT SIMPLE: ";: INVERSE
     : LIST 10050: NORMAL
10050  &  INPUT A$
10060  PRINT "AVEC LABEL: ";: INVERSE
     : LIST 10070: NORMAL
10070  &  INPUT B$,"ESSAI   "
10080  PRINT "AVEC FORMAT: ";: INVERSE
     : LIST 10090: NORMAL
10090  &  INPUT C$,L20
10100  PRINT "FORMAT,TABULATION,LA
     BEL:";: INVERSE : LIST 10110:
      NORMAL
10110  &  INPUT D2$,L100,V24,H4,"D
     EMONSTRATION:"
10120  PRINT "PASSAGE DE PARAMETRE
     S:";
10130  INVERSE : LIST 10140,10150:
      NORMAL
10140 A = 10:B = 24:C = 4
10150  &  INPUT E$,HA,VB,LC,"DEMO:
     "
10151  PRINT
10160  HOME
10161  HTAB 8: PRINT "** ACCES DIS
     QUE AMELIORE **": PRINT
10165 F$ = "PROGR DEMO TEMP"
10170 D$ =  CHR$ (4): PRINT D$"MON
     C,I,O"
10171  PRINT D$"OPEN "F$
10172  PRINT D$"WRITE"F$
10173  PRINT A$: PRINT B$: PRINT C
     $: PRINT D2$: PRINT E$
10174  PRINT D$"CLOSE"
10175  PRINT D$"OPEN"F$
10176  PRINT D$"READ"F$
10177  FOR I = 1 TO 5
10178  &  INPUT A$(I): NEXT I
10179  PRINT D$"CLOSE"
10220  &  WAIT
10222  REM DANS L'INSTRUCTION
10223  REM 'FORMAT' LA VARIABLE ES
     T
10224  REM EN PREMIERE POSITION
10225  REM LE LABEL EN DERNIERE
10226  REM TANDIS QUE L'ORDRE DES
10227  REM PARAMETRES NUMERIQUES
10228  REM EST LIBRE
10235  HOME
19990  REM ----------------------
          RECAPITULATION DES
        NOUVELLES COMMANDES
        ----------------------
20000  TEXT : HOME
20010  PRINT "  R E C A P I T U L
     A T I F"
20015  PRINT "  -FORMAT DES INSTRU
     CTIONS-"
20020  PRINT "--------------------
     --------------------"
20040  PRINT "INPUT AVEC FORMAT:":
      PRINT
20050  PRINT " & INPUT A$,L<LONGUE
     UR>,V<VTAB>,H<HTAB>," CHR$ (3
```

```
     4)"TEXTE" CHR$ (34)
20060  PRINT : PRINT "MENU:": PRINT

20070  PRINT " & LET A%,<VTAB>,<MA
     RGE>,<LARG.>,<NBR>"
20080  PRINT : PRINT "AMPER GO:": PRINT

20090  PRINT " & GOTO EXPRESSION"
20100  PRINT " & GOSUB EXPRESSION"

20110  PRINT : PRINT "ATTENTE:": PRINT

20120  PRINT " & WAIT"
```

```
                 1     ********************
                 2     *------------------*
                 3     *                  *
                 4     *   PROGRAMMATION  *
                 5     *     FACILITEE    *
                 6     *       ----       *
                 7     * PAR DENIS SUREAU *
                 8     *                  *
                 9     *------------------*
                 10    ********************
                 11
                 12    ********************
                 13    **                **
                 14    **   F O R M A T  **
                 15    **                **
                 16    ********************
                 17
                 18    FORPNT   EQU  $0085
                 19    CHRGET   EQU  $00B1
                 20    TXTPTR   EQU  $00B8
                 21    GETBYTC  EQU  $E6F5
                 22    FRMNUM   EQU  $DD67
                 23    PTRGET   EQU  $DFE3
                 24    PRINT    EQU  $FDED
                 25    BS       EQU  $FC10
                 26    VTAB     EQU  $FC22
                 27    CHKSTR   EQU  $DD6C
                 28    MAK      EQU  $E3E9
                 29    SAVD     EQU  $DA9A
                 30    SETNORM  EQU  $FE84
                 31
                 32    LONGUEUR EQU  $0009
                 33
                 34             ORG  $9300
                 35
9300: A9 4C      36    INIT     LDA  #$4C
9302: 8D F5 03   37             STA  $3F5
9305: A9 1B      38             LDA  #<SELECT
9307: 8D F6 03   39             STA  $3F6
930A: A9 93      40             LDA  #>SELECT
930C: 8D F7 03   41             STA  $3F7
930F: A9 00      42             LDA  #<INIT
9311: 85 73      43             STA  $73
9313: A9 93      44             LDA  #>INIT
9315: 85 74      45             STA  $74
```

```
9317: 60          46              RTS
                  47
9318: 02          48   INTERLN    HEX  02
9319: BE          49   FLECHE     ASC  ">"
931A: 80          50   DUREE      HEX  80
                  51
931B: C9 84       52   SELECT     CMP  #$84        *SELECTION
931D: F0 1D       53              BEQ  INPUT       *DE LA
931F: C9 AB       54              CMP  #$AB        *ROUTINE
9321: F0 13       55              BEQ  GO
9323: C9 B0       56              CMP  #$B0
9325: F0 12       57              BEQ  SUB
9327: C9 B5       58              CMP  #$B5
9329: F0 08       59              BEQ  WT
932B: C9 AA       60              CMP  #$AA
932D: D0 03       61              BNE  RTSSEL
932F: 4C 4E 94    62              JMP  MENU
9332: 60          63   RTSSEL     RTS
9333: 4C 2F 94    64   WT         JMP  WAIT
9336: 4C FF 93    65   GO         JMP  AMPGO
9339: 4C 05 94    66   SUB        JMP  AMPSUB
                  67
933C: A4 25       68   INPUT      LDY  $25
933E: 84 07       69              STY  $7
9340: A4 24       70              LDY  $24
9342: 84 08       71              STY  $8
9344: A9 00       72              LDA  #$00
9346: 85 09       73              STA  LONGUEUR
9348: 85 06       74              STA  $6
934A: 20 B1 00    75              JSR  CHRGET      *RECHERCHE
934D: 20 E3 DF    76              JSR  PTRGET      *DU NOM DE
9350: 20 6C DD    77              JSR  CHKSTR      *VARIABLE
9353: 85 85       78              STA  FORPNT      *DANS LA
9355: 84 86       79              STY  FORPNT+1    *TABLE
                  80
9357: A0 00       81   I2         LDY  #$00        *ENTREE
9359: B1 B8       82              LDA  (TXTPTR),Y  *DES
935B: C9 2C       83              CMP  #$2C        *PARAMETRES
935D: D0 35       84              BNE  TAB
935F: 20 B1 00    85              JSR  CHRGET
9362: C9 4C       86              CMP  #$4C
9364: F0 14       87              BEQ  INPTL
9366: C9 56       88              CMP  #$56
9368: F0 18       89              BEQ  INPTV
936A: C9 48       90              CMP  #$48
936C: F0 1D       91              BEQ  INPTH
936E: C9 22       92              CMP  #$22
9370: D0 05       93              BNE  JMPERR
9372: E6 06       94              INC  $6
9374: 4C 94 93    95              JMP  TAB
9377: 4C 8D 94    96   JMPERR     JMP  ERR
937A: 20 F5 E6    97   INPTL      JSR  GETBYTC
937D: 86 09       98              STX  LONGUEUR
937F: 4C 57 93    99              JMP  I2
9382: 20 F5 E6    100  INPTV      JSR  GETBYTC
9385: CA          101             DEX
9386: 86 07       102             STX  $7
9388: 4C 57 93    103             JMP  I2
938B: 20 F5 E6    104  INPTH      JSR  GETBYTC
938E: CA          105             DEX
```

```
938F: 86 08      106            STX  $8
9391: 4C 57 93   107            JMP  I2
9394: A5 07      108  TAB       LDA  $7          *POSITION
9396: 85 25      109            STA  $25         *VERTICALE
9398: A5 08      110            LDA  $8          *EN $25
939A: 85 24      111            STA  $24         *ET EN $24
939C: 20 22 FC   112            JSR  VTAB        *HORIZONTALE
939F: A5 06      113            LDA  $6
93A1: F0 12      114            BEQ  FORMAT
93A3: 20 B1 00   115  STRING    JSR  CHRGET      *AFFICHAGE
93A6: C9 22      116            CMP  #$22        *DU
93A8: F0 08      117            BEQ  RTSSTR      *LABEL
93AA: 09 80      118            ORA  #$80
93AC: 20 ED FD   119            JSR  PRINT
93AF: 4C A3 93   120            JMP  STRING
93B2: 20 B1 00   121  RTSSTR    JSR  CHRGET
93B5: A6 09      122  FORMAT    LDX  LONGUEUR    *ECRITURE
93B7: F0 19      123            BEQ  GTLIN       *D'UNE
93B9: A5 32      124            LDA  $32         *LIGNE
93BB: 48         125            PHA  *DE
93BC: 20 84 FE   126            JSR  SETNORM     *POINTS
93BF: A9 AE      127            LDA  #"."
93C1: 20 ED FD   128  FORMT2    JSR  PRINT
93C4: CA         129            DEX
93C5: D0 FA      130            BNE  FORMT2
93C7: A6 09      131            LDX  LONGUEUR    *PUIS
93C9: 20 10 FC   132  FORMT3    JSR  BS          *RETOUR
93CC: CA         133            DEX  *EN
93CD: D0 FA      134            BNE  FORMT3      *POSITION
93CF: 68         135            PLA  *INITIALE
93D0: 85 32      136            STA  $32
93D2: 20 6F F0   137  GTLIN     JSR  $FD6F       *ENTREE
93D5: 8A         138            TXA  *DES
93D6: F0 1E      139            BEQ  GTRTS       *DONNEES
93D8: BD FF 01   140            LDA  $1FF,X
93D8: C9 83      141            CMP  #$83        *CTRL-C?
93DD: F0 1A      142            BEQ  STOP
                 143
93DF: BD FF 01   144  GTLIN2    LDA  $1FF,X      *ON MET
93E2: 29 7F      145            AND  #$7F        *LE BIT
93E4: 9D FF 01   146            STA  $1FF,X      *D'ORDRE
93E7: CA         147            DEX  *7          A 0
93E8: D0 F5      148            BNE  GTLIN2      *ZERO
93EA: A9 00      149  AFFECTE   LDA  #$00
93EC: A0 02      150            LDY  #$02        *TRANSFERT
93EE: A2 8D      151            LDX  #$8D        *DU BUFFER
93F0: 20 E9 E3   152            JSR  MAK         *D'ENTREE
93F3: 20 9A DA   153            JSR  SAVD        *VERS LA
93F6: A2 00      154  GTRTS     LDX  #$00        MEMOIRE
93F8: 60         155            RTS
93F9: 20 3A FF   156  STOP      JSR  $FF3A       *BIP
93FC: 4C D0 03   157            JMP  $3D0
                 158
                 159  ********************
                 160  **                **
                 161  ** A M P E R  G O **
                 162  **                **
                 163  ********************
                 164
                 165  GETADR    EQU  $E752
```

```
                 166  GOTO     EQU  $D941
                 167  CHKSTK   EQU  $D3D6
                 168  CURLIN   EQU  $0075
                 169  NEWSTT   EQU  $D7D2
                 170  *
93FF: 20 25 94   171  AMPGO    JSR  AMPRTN
9402: 4C 41 D9   172           JMP  GOTO
9405: 20 25 94   173  AMPSUB   JSR  AMPRTN
9408: A9 03      174           LDA  #$03
940A: 20 D6 D3   175           JSR  CHKSTK
940D: A5 B9      176           LDA  TXTPTR+1
940F: 48         177           PHA
9410: A5 B8      178           LDA  TXTPTR
9412: 48         179           PHA
9413: A5 76      180           LDA  CURLIN+1
9415: 48         181           PHA
9416: A5 75      182           LDA  CURLIN
9418: 48         183           PHA
9419: A9 B0      184           LDA  #$B0
941B: 48         185           PHA
941C: 20 B7 00   186           JSR  $B7
941F: 20 41 D9   187           JSR  GOTO
9422: 4C D2 D7   188           JMP  NEWSTT
9425: 20 B1 00   189  AMPRTN   JSR  CHRGET      *EVALUATION
9428: 20 67 DD   190           JSR  FRMNUM      *DE LA
942B: 20 52 E7   191           JSR  GETADR      *FORMULE
942E: 60         192           RTS
                 193
                 194  **********************
                 195  **     W A I T     **
                 196  **********************
                 197
942F: 20 B1 00   198  WAIT     JSR  CHRGET
9432: A2 FF      199           LDX  #$FF
9434: 2C 10 C0   200           BIT  $C010
9437: CA         201  WAIT2    DEX
9438: F0 13      202           BEQ  RTSWT
943A: AC 1A 93   203           LDY  DUREE
943D: 88         204  WAIT3    DEY
943E: F0 F7      205           BEQ  WAIT2
                 206  *$FCA8 EST LA BOUCLE D'ATTENTE
                 207  *DU MONITEUR,FONCTION DE A
9440: A9 0A      208           LDA  #10
9442: 20 A8 FC   209           JSR  $FCA8
9445: 2C 00 C0   210           BIT  $C000       *UNE
9448: 10 F3      211           BPL  WAIT3       *TOUCHE
944A: 2C 10 C0   212           BIT  $C010       *PRESSEE?
944D: 60         213  RTSWT    RTS
                 214
                 215  **********************
                 216  **                  **
                 217  **S E L E C T I O N**
                 218  **                  **
                 219  **********************
                 220
                 221  KBD      EQU  $C000
                 222  KBDSTRB  EQU  $C010
                 223  VARPNT   EQU  $0083
                 224  CHKNUM   EQU  $DD6A
                 225  HAUT     EQU  $0008       *LIGNE 1
```

```
                226
                227  *NUMOPT: NUMERO DE L'OPTION
                228  *        OU SE TROUVE LE CURSEUR
                229  NUMOPT   EQU  $0009
                230  NOPT     EQU  $000B       *NBR D'OPT.
                231  NOPT2    EQU  $000C
                232  SETINV   EQU  $FE80
                233  PRERR    EQU  $FF2D
                234  CV       EQU  $0025
                235
                236  ** DEFINITION DE LA VARIABLE
                237
944E: 20 B1 00  238  MENU     JSR  CHRGET
9451: 20 E3 DF  239          JSR  PTRGET
9454: 48        240          PHA  *SAUVEG.
9455: 98        241          TYA  *DE
9456: 48        242          PHA  *L'ADRESSE
                243
                244  *** ENTREE DES PARAMETRES
                245
9457: 20 F5 E6  246          JSR  GETBYTC     *VTAB
945A: CA        247          DEX
945B: E0 18     248          CPX  #24
945D: B0 2E     249          BCS  ERR
945F: 86 08     250          STX  HAUT
9461: 20 F5 E6  251          JSR  GETBYTC     *MARGE
9464: 8A        252          TXA
9465: F0 26     253          BEQ  ERR
9467: CA        254          DEX
9468: 86 06     255          STX  $6
946A: 20 F5 E6  256          JSR  GETBYTC     *LARG.
946D: 8A        257          TXA
946E: 18        258          CLC
946F: 65 06     259          ADC  $6
9471: C9 29     260          CMP  #41
9473: B0 18     261          BCS  ERR
9475: 85 07     262          STA  $7
9477: 20 F5 E6  263          JSR  GETBYTC     *NBR OP.
947A: 86 0B     264          STX  NOPT
947C: E8        265          INX
947D: 8A        266          TXA
947E: 09 B0     267          ORA  #$B0
9480: 85 0C     268          STA  NOPT2
9482: A9 01     269  PREMOPT LDA  #$1         *INITIAL.
9484: 85 09     270          STA  NUMOPT
9486: A5 08     271          LDA  HAUT
9488: 85 25     272          STA  CV
948A: 4C C3 94  273          JMP  BARRINV
                274
948D: 20 84 FE  275  ERR      JSR  SETNORM
9490: 20 2D FF  276          JSR  PRERR       *AFFICHE
9493: 4C D0 03  277          JMP  $3D0        *'ERR'
                278  *
9496: 00        279  FLECHEZ  HEX  00
                280
                281  *** MONTEE DU CURSEUR
                282
9497: 20 15 95  283  MONTRTN  JSR  BARRNORM
949A: A5 09     284          LDA  NUMOPT
949C: C9 02     285          CMP  #$2
```

```
949E: B0 15      286            BCS  MONTRT2
94A0: A5 08      287            LDA  HAUT
94A2: A4 0B      288            LDY  NOPT
94A4: 88         289            DEY
94A5: 18         290  MNTLOOP   CLC  *SAUT
94A6: 6D 18 93   291            ADC  INTERLN     *EN
94A9: 88         292            DEY  *DERNIERE
94AA: D0 F9      293            BNE  MNTLOOP     *LIGNE
94AC: 85 25      294            STA  CV
94AE: A5 0B      295            LDA  NOPT
94B0: 85 09      296            STA  NUMOPT
94B2: 4C C3 94   297            JMP  BARRINV
94B5: 20 15 95   298  MONTRT2   JSR  BARRNORM
94B8: A5 25      299            LDA  CV
94BA: 69 00      300            ADC  #$0
94BC: ED 18 93   301            SBC  INTERLN
94BF: 85 25      302            STA  CV
94C1: C6 09      303            DEC  NUMOPT
94C3: 20 80 FE   304  BARRINV   JSR  SETINV
94C6: AD 19 93   305            LDA  FLECHE
94C9: 8D 96 94   306            STA  FLECHE2
94CC: 20 43 95   307            JSR  BARR
94CF: 2C 10 C0   308  CLAV      BIT  KBDSTRB     *ATTENTE
94D2: 2C 00 C0   309  BCLCLAV   BIT  KBD         *DU
94D5: 10 FB      310            BPL  BCLCLAV     *CLAVIER
94D7: AD 00 C0   311            LDA  KBD
94DA: 2C 10 C0   312            BIT  KBDSTRB
94DD: C9 95      313            CMP  #$95        *->?
94DF: F0 40      314            BEQ  DESRTN
94E1: C9 88      315            CMP  #$88        *<-?
94E3: F0 B2      316            BEQ  MONTRTN
94E5: C9 9B      317            CMP  #$9B        *ESC?
94E7: F0 51      318            BEQ  ESCAPE
94E9: C9 8D      319            CMP  #$8D        *RTN?
94EB: F0 0C      320            BEQ  AFFECTE2
94ED: C9 B1      321            CMP  #$B1        *UN
94EF: 90 E1      322            BCC  BCLCLAV     *CHIFFRE?
94F1: C5 0C      323            CMP  NOPT2
94F3: B0 DD      324            BCS  BCLCLAV
94F5: 29 0F      325            AND  #$F
94F7: 85 09      326            STA  NUMOPT
                 327
                 328  *** TRANSFERT DU NUMERO
                 329  *** DANS LA VARIABLE
                 330
94F9: 68         331  AFFECTE2  PLA
94FA: 85 83      332            STA  VARPNT
94FC: 68         333            PLA
94FD: 85 84      334            STA  VARPNT+1
                 335
                 336  GIVAYF    EQU  $E2F2
                 337  MOVMF     EQU  $EB2B
                 338
94FF: A9 00      339            LDA  #$00
9501: A4 09      340            LDY  NUMOPT
9503: 20 F2 E2   341            JSR  GIVAYF      *INT>FAC
9506: A6 84      342            LDX  VARPNT+1
9508: A4 83      343            LDY  VARPNT
950A: 20 2B EB   344            JSR  MOVMF       *FAC>MEM
950D: A9 00      345            LDA  #$0
```

```
950F: 85 24     346            STA  $24
9511: 20 84 FE  347  RTS       JSR  SETNORM
9514: 60        348            RTS
9515: 20 84 FE  349  BARRNORM  JSR  SETNORM
9518: A9 A0     350            LDA  #$A0
951A: 8D 96 94  351            STA  FLECHE2
951D: 20 50 95  352            JSR  BARR3
9520: 60        353            RTS
                354
                355  *** DESCENTE DU CURSEUR
                356
9521: 20 15 95  357  DESRTN    JSR  BARRNORM
9524: A5 09     358            LDA  NUMOPT
9526: C5 0B     359            CMP  NOPT
9528: 90 03     360            BCC  DESRT2
952A: 4C 82 94  361            JMP  PREMOPT
952D: A5 25     362  DESRT2    LDA  CV
952F: 18        363            CLC
9530: 6D 18 93  364            ADC  INTERLN
9533: 85 25     365            STA  CV
9535: E6 09     366            INC  NUMOPT
9537: 4C C3 94  367            JMP  BARRINV
953A: 20 58 FC  368  ESCAPE    JSR  $FC58       *ARRET
953D: 20 84 FE  369            JSR  SETNORM
9540: 4C D0 03  370            JMP  $3D0
9543: A5 25     371  BARR      LDA  CV
9545: C9 18     372            CMP  #24
9547: 90 04     373            BCC  BARR2
9549: A9 17     374            LDA  #23
954B: 85 25     375            STA  CV
954D: 20 22 FC  376  BARR2     JSR  VTAB
9550: A4 06     377  BARR3     LDY  $6
9552: C4 07     378            CPY  $7
9554: D0 06     379            BNE  BARRWRT
9556: AD 96 94  380            LDA  FLECHE2
9559: 91 28     381            STA  ($28),Y
955B: 60        382  RTSBARR   RTS
955C: B1 28     383  BARRWRT   LDA  ($28),Y     *REECRITURE
955E: 09 80     384            ORA  #$80        *EN
9560: 25 32     385            AND  $32         *INVERSE
9562: 91 28     386            STA  ($28),Y     *OU EN
9564: C8        387            INY  *NORMAL
9565: C4 07     388            CPY  $7
9567: 90 F3     389            BCC  BARRWRT
9569: 60        390            RTS
```

Récapitulation 48K

```
9300- A9 4C 8D F5 03 A9 1B 8D
9308- F6 03 A9 93 8D F7 03 A9
9310- 00 85 73 A9 93 85 74 60
9318- 02 BE 80 C9 84 F0 1D C9
9320- AB F0 13 C9 B0 F0 12 C9
9328- B5 F0 08 C9 AA D0 03 4C
9330- 4E 94 60 4C 2F 94 4C FF
9338- 93 4C 05 94 A4 25 84 07
9340- A4 24 84 08 A9 00 85 09
9348- 85 06 20 B1 00 20 E3 DF
9350- 20 6C DD 85 85 84 86 A0
9358- 00 B1 B8 C9 2C D0 35 20
9360- B1 00 C9 4C F0 14 C9 56
9368- F0 18 C9 48 F0 1D C9 22
9370- D0 05 E6 06 4C 94 93 4C
9378- 8D 94 20 F5 E6 86 09 4C
9380- 57 93 20 F5 E6 CA 86 07
9388- 4C 57 93 20 F5 E6 CA 86
9390- 08 4C 57 93 A5 07 85 25
9398- A5 08 85 24 20 22 FC A5
93A0- 06 F0 12 20 B1 00 C9 22
93A8- F0 08 09 80 20 ED FD 4C
```

```
93B0- A3 93 20 B1 00 A6 09 F0
93B8- 19 A5 32 48 20 84 FE A9
93C0- AE 20 ED FD CA D0 FA A6
93C8- 09 20 10 FC CA D0 FA 68
93D0- 85 32 20 6F FD 8A F0 1E
93D8- BD FF 01 C9 83 F0 1A BD
93E0- FF 01 29 7F 9D FF 01 CA
93E8- D0 F5 A9 00 A0 02 A2 8D
93F0- 20 E9 E3 20 9A DA A2 00
93F8- 60 20 3A FF 4C D0 03 20
9400- 25 94 4C 41 D9 20 25 94
9408- A9 03 20 D6 D3 A5 B9 48
9410- A5 B8 48 A5 76 48 A5 75
9418- 48 A9 B0 48 20 B7 00 20
9420- 41 D9 4C D2 D7 20 B1 00
9428- 20 67 DD 20 52 E7 60 20
9430- B1 00 A2 FF 2C 10 C0 CA
9438- F0 13 AC 1A 93 88 F0 F7
9440- A9 0A 20 A8 FC 2C 00 C0
9448- 10 F3 2C 10 C0 60 20 B1
9450- 00 20 E3 DF 48 98 48 20
9458- F5 E6 CA E0 18 B0 2E 86
9460- 08 20 F5 E6 8A F0 26 CA
9468- B6 06 20 F5 E6 8A 18 65
9470- 06 C9 29 B0 18 85 07 20
9478- F5 E6 86 0B E8 8A 09 B0
9480- 85 0C A9 01 85 09 A5 08
9488- 85 25 4C C3 94 20 84 FE
9490- 20 2D FF 4C D0 03 00 20
9498- 15 95 A5 09 C9 02 B0 15
94A0- A5 08 A4 0B 88 18 6D 18
94A8- 93 88 D0 F9 85 25 A5 0B
94B0- 85 09 4C C3 94 20 15 95
94B8- A5 25 69 00 ED 18 93 85
94C0- 25 C6 09 20 80 FE AD 19
94C8- 93 8D 96 94 20 43 95 2C
94D0- 10 C0 2C 00 C0 10 FB AD
94D8- 00 C0 2C 10 C0 C9 95 F0
94E0- 40 C9 88 F0 B2 C9 9B F0
94E8- 51 C9 8D F0 0C C9 B1 90
94F0- E1 C5 0C B0 DD 29 0F 85
94F8- 09 68 85 83 68 85 84 A9
9500- 00 A4 09 20 F2 E2 A6 84
9508- A4 83 20 2B EB A9 00 85
9510- 24 20 B4 FE 60 20 84 FE
9518- A9 A0 8D 96 94 20 50 95
9520- 60 20 15 95 A5 09 C5 0B
9528- 90 03 4C 82 94 A5 25 18
9530- 6D 18 93 85 25 E6 09 4C
9538- C3 94 20 58 FC 20 84 FE
9540- 4C D0 03 A5 25 C9 18 90
9548- 04 A9 17 85 25 20 22 FC
9550- A4 06 C4 07 D0 06 AD 96
9558- 94 91 28 60 B1 28 09 B0
9560- 25 32 91 28 C8 C4 07 90
9568- F3 60
```

RECAPITULATION HEXADECIMALE : VERSION 32K

```
5300- A9 4C 8D F5 03 A9 1B 8D
5308- F6 03 A9 53 8D F7 03 A9
5310- 00 85 73 A9 53 85 74 60
5318- 02 BE 80 C9 84 F0 1D C9
5320- AB F0 13 C9 B0 F0 12 C9
5328- B5 F0 0B C9 AA D0 03 4C
5330- 4E 54 60 4C 2F 54 4C FF
5338- 53 4C 05 54 A4 25 84 07
5340- A4 24 84 08 A9 00 85 09
5348- 85 06 20 B1 00 20 E3 DF
5350- 20 6C DD 85 85 84 86 A0
5358- 00 B1 B8 C9 2C D0 35 20
5360- B1 00 C9 4C F0 14 C9 56
5360- F0 18 C9 4B F0 1D C9 22
5370- D0 05 E6 06 4C 94 53 4C
5378- 8D 54 20 F5 E6 86 09 4C
5380- 57 53 20 F5 E6 CA 86 07
5388- 4C 57 53 20 F5 E6 CA 86
5390- 08 4C 57 53 A5 07 85 25
5398- A5 08 85 24 20 22 FC A5
53A0- 06 F0 12 20 B1 00 C9 22
53A8- F0 08 09 80 20 ED FD 4C
53B0- A3 53 20 B1 00 A6 09 F0
53B8- 19 A5 32 48 20 84 FE A9
53C0- AE 20 ED FD CA D0 FA A6
53C8- 09 20 10 FC CA D0 FA 68
53D0- 85 32 20 6F FD 8A F0 1E
53D8- BD FF 01 C9 83 F0 1A BD
53E0- FF 01 29 7F 9D FF 01 CA
53E8- D0 F5 A9 00 A0 02 A2 8D
53F0- 20 E9 E3 20 9A DA A2 00
53F8- 60 20 3A FF 4C D0 03 20
5400- 25 54 4C 41 D9 20 25 54
5408- A9 03 20 D6 D3 A5 B9 48
5410- A5 B8 48 A5 76 48 A5 75
5418- 48 A9 B0 48 20 B7 00 20
5420- 41 D9 4C D2 D7 20 B1 00
5428- 20 67 DD 20 52 E7 60 20
5430- B1 00 A2 FF 2C 10 C0 CA
5438- F0 13 AC 1A 53 88 F0 F7
5440- A9 0A 20 A8 FC 2C 00 C0
5448- 10 F3 2C 10 C0 60 20 B1
5450- 00 20 E3 DF 48 98 48 20
5458- F5 E6 CA E0 18 B0 2E 86
5460- 08 20 F5 E6 8A F0 26 CA
5468- 86 06 20 F5 E6 8A 18 65
5470- 06 C9 29 B0 18 85 07 20
5478- F5 E6 86 0B E8 8A 09 B0
5480- 85 0C A9 01 85 09 A5 08
5488- 85 25 4C C3 54 20 84 FE
5490- 20 2D FF 4C D0 03 00 20
5498- 15 55 A5 09 C9 02 B0 15
54A0- A5 08 A4 0B 88 18 6D 18
54A8- 53 00 D0 F9 85 25 A5 0B
54B0- 85 09 4C C3 54 20 15 55
54B8- A5 25 69 00 ED 18 53 85
54C0- 25 C6 09 20 80 FE AD 19
54C8- 53 8D 96 54 20 43 55 2C
54D0- 10 C0 2C 00 C0 10 FB AD
54D8- 00 C0 2C 10 C0 C9 95 F0
54E0- 40 C9 88 F0 B2 C9 9B F0
54E8- 51 C9 8D F0 0C C9 B1 90
54F0- E1 C5 0C B0 DD 29 0F 85
54F8- 09 68 85 83 68 85 84 A9
5500- 00 A4 09 20 F2 E2 A6 84
5508- A4 83 20 2B EB A9 00 85
5510- 24 20 84 FE 60 20 84 FE
5518- A9 A0 8D 96 54 20 50 55
5520- 60 20 15 55 A5 09 C5 0B
5528- 90 03 4C 82 54 A5 25 18
5530- 6D 18 53 85 25 E6 09 4C
5538- C3 54 20 58 FC 20 84 FE
5540- 4C D0 03 A5 25 C9 18 90
5548- 04 A9 17 85 25 20 22 FC
5550- A4 06 C4 07 D0 06 AD 96
5558- 54 91 28 60 B1 28 09 80
5560- 25 32 91 28 C8 C4 07 90
5568- F3 60
```

Banc-test de la carte Legend 128K DE

Olivier Herz

La carte 128KDE de Legend Industries LTD. est le résultat des technologies les plus récentes: sur la taille d'une carte langage, on peut maintenant faire tenir 128K. Auparavant, Legend ne fournissait que des cartes 64K. Bien entendu, le microprocesseur 6502 de l'Apple II ne peut adresser lui-même les 128K de la carte, mais seulement un morceau de 16K (adresses $D000 à $FFFF). Pour changer la partie adressée, il suffira de stimuler des adresses (faire un SWITCH).

Cette carte peut aussi bien être placee dans le slot 0 à la place de la carte langage ou d'une RAMCARD de 16K que dans n'importe quel autre slot, à l'exception du slot 7 (réservé pour les cartes couleur) pour les Apples Europlus.

Le hardware de cette carte est relativement aisé à comprendre: il s'agit de l'équivalent de 8 cartes langage en parallèle et le passage de l'une à l'autre s'effectue par un SWITCH. En effet, la carte est formée de 8 blocs (banks) de 16K. Une fois qu'un bloc est sélectionné, on peut l'adresser comme une carte langage: il possède lui-même 2 banks allant des adresses $D000 à $DFFF et une partie commune de $E000 à $FFFF, et l'on peut accéder à l'écriture ou à la lecture de ces adresses en stimulant les mêmes adresses que pour une RAMCARD ($C080 à $C084 et $C088 à $C08B pour le slot 0 - ajouter $i0 pour le slot numéro i). Puis, pour passer d'un bloc de 16K à l'autre il suffit de mettre dans l'adresse $C084 + $i0 le numéro du bloc (de 0 à 7). Lorsque la carte est placée dans le slot 0, le bloc 0 est l'équivalent de la carte langage: c'est sur lui que sera chargé le langage qui ne réside pas en ROM.

Cette carte serait moyennement utile si elle n'était munie d'aucun logiciel, mais Legend a développé pour cette carte un logiciel d'un très grand interêt. Ce logiciel est principalement constitué de 3 programmes: VC PLUS, DISK EMULATOR et MEMORY MASTER. On peut utiliser ces programmes de deux manières différentes: soit en les BRUNnant (dans ce cas ils demandent à l'utilisateur les paramètres dont ils ont besoin), soit dans un programme automatique (TURNKEY) qui peut être utilisé en bootant (dans ce cas un programme Applesoft fait des POKEs dans la routine en langage machine et termine par un CALL). Notons que ces logiciels sont prévus pour utiliser jusqu'à 4 cartes Legend simultanément et qu'ils sont incompatibles entre eux.

VC PLUS permet d'adapter Visicalc aux cartes 128K. La présence d'une carte porte l'espace de travail à 145K; il est de 85K avec la carte 64K aussi commercialisée par la même société.

Le programme DISK EMULATOR permet de simuler un drive "fantôme" (dont les numéros de slot et drive peuvent être quelconques, à condition de ne pas correspondre à un drive existant, qui sinon serait déconnecté) en utilisant la carte Legend comme mémoire de masse. En effet, le contenu d'une disquette est exactement de 128K (pistes 3 à 34), si l'on excepte le DOS qui se trouve sur les pistes 0, 1 et 2. Le DOS possède alors deux commandes supplémentaires qui agissent en moins de 20 secondes:

.Mi,Ss,Dd (MOUNT) recopie le contenu du drive d, slot s dans la carte 128K du slot i
.Ui,Ss,Dd (UPDATE) recopie le contenu de la carte dans le drive.

Notons que si la carte est située dans le slot 0 et que l'on désire réserver le premier bloc de la carte pour le langage non résident en ROM, seules les pistes 7 à 34 du drive fantôme seront utilisées. Pour pouvoir dans ce cas se servir des commandes MOUNT et UPDATE, il faut formater des disquettes qui n'utilisent pour le stockage des données que les pistes 7 à 34. Cela peut être réalisé avec le programme SPECIAL FORMATTER, qui en fait ne consiste qu'à faire avant initialisation POKE-20813, 4*(piste du début) et POKE-20811,4*(1+piste de la fin).

L'utilisation de l'émulateur de disque permet de charger presque instantanément les plus gros programmes et de diviser par 3 le temps d'écriture ou de lecture des fichiers TEXT. Mais il ne faut pas perdre de vue que le disque émulé reste une mémoire volatile et qu'il faut donc souvent sauvegarder le fruit de son travail.

Le programme MEMORY MASTER permet de recopier la majorité du DOS dans un des blocs de 16K de la carte Legend. La HIMEM est alors portée à $B800, ce qui laisse 44K dans la mémoire centrale pour écrire des programmes. Notons que MEMORY MASTER peut être utilisé par les Apples munis simplement de la carte langage (à ce moment-là, on ne peut plus utiliser le langage en RAM). Ce programme reste compatible avec tous ceux qui se contentent d'appels au DOS par l'intermédiaire des commandes DOS ou des

adresses de la page 3.

Les commandes suivantes sont alors ajoutées au DOS:

.F (FLIP) permet de passer de DOS 3.3 en DOS 3.2 et vice-versa (sans effacer le programme en mémoire!)
.S (SHOW) montre la version de DOS utilisée
.B (BSTAT) fournit l'adresse hexadécimale du début et la longueur du dernier fichier BLOADé ou BRUNné
.M (MONITOR) fait entrer dans le moniteur.

Signalons enfin les programmes FIRMWARE SELECT qui modifie le DOS sur une disquette pour lui indiquer la présence dans un slot donné d'une ROMCARD Integer ou Applesoft, HIRES DEMO qui fait une démonstration par chargement d'images graphiques de la vitesse de la carte 128K comparée au disque et Legend SLIDE SELECT qui simule un projecteur de diapositives.

Un mot à propos du DOS booté par la disquette 128KDE DEMO: il charge très rapidement les programmes binaires et Basic, à la manière du & BLOAD de Pom's 4. De plus il charge automatiquement sur la carte langage ou sur le bloc 0 d'une carte Legend située dans le slot 0 le langage non résident en ROM (à moins que celui-ci ne soit déjà chargé). D'ailleurs, l'Integer et l'Applesoft ne sont pas fournis par des fichiers mais sont directement cherchés par le DOS booté sur les pistes 3 à 7 de la disquette, pistes sur lesquelles on ne peut pas sauver des fichiers. Ce DOS peut être utilisé par initialisation ou être recopié sur les pistes 0 à 2 d'une disquette comme indiqué par J.F. Duvivier dans Pom's 2 (MASTER CREATE ne marche pas).

Les produits Legend sont distribués en France par B.I.P. (22, rue Joseph Dijon - 75018 Paris) à qui nous devons l'excellente idée d'une traduction en français du mode d'emploi américain qui est d'ailleurs très clair (sauf en ce qui concerne l'installation d'une carte ailleurs que dans le slot 0). La traduction n'est malheureusement pas parfaite: elle a été, comme l'original, rédigée sur un système de traitement de texte et on peut s'étonner de l'absence des accents (alors que la plupart des imprimantes en ont - cf. les articles de Pom's) et de l'absence du mode de justification à droite et à gauche (il n' y a qu'une justification à gauche). Depuis peu, la société BIP propose la documentation sur disquette en AppleWriter I ou II, ce qui permet à chacun d'avoir sa documentation personnalisée et agrémentée d'accents ....

Sur le plan du fond, la traduction est trop littérale ("the system hangs" devient "le système est suspendu" - "planté" serait meilleur; "you must LOAD the program" devient "vous devez "LOAD" le programme" - le barbarisme LOADer serait plus clair; "you re-boot DOS" donne "vous redémarrez DOS" - on dit habituellement: le DOS). Notons toutefois que, tout au long de la traduction, l'accent est mis sur le fait qu'avec les Apples Europlus on ne peut utiliser que les slots 0 à 6, alors que le mode d'emploi américain parle des slots 0 à 7.

En conclusion, il s'agit là d'un produit qui peut se révéler d'une très grande utilité grâce à l'existence d'un excellent logiciel. Mais il faudrait adapter pour cette carte d'autres programmes que Visicalc. Par exemple, il serait intéressant d'avoir une adaptation du système de traitement de texte Apple Writer. La seule réserve que nous pouvons émettre est que le prix de 5600F H.T. est un peu élevé, mais c'est dû au prix encore important de la mémoire vive. Rappelons à titre de comparaison que les 16K de RAM pour Apple coûtent environ 500F!

Nous avons beaucoup aimé la carte 128K en Visicalc, ainsi que pour le chargement et l'utilisation de programmes en DOS normal. Nous avons regretté de ne pas pouvoir l'utiliser avec plus de programmes du commerce. Dès que le DOS est modifié, l'utilisateur doit être assez compétent pour réaliser les "patches" nécessaires ...

LIBRAIRIE INFORMATIQUE
LA NACELLE
ÉLECTRONIQUE • AUTOMATISME • MICROPROCESSEUR
TOUS OUVRAGES ET ABONNEMENTS
FRANÇAIS ET ETRANGERS
Distributeur exclusif pour la France des manuels techniques du Réseau Calvados
Tous les ouvrages français ou étrangers signalés dans cette revue peuvent être obtenus ou commandés à La Nacelle
2, rue Campagne-Première 75014 PARIS - Tél. 322 56 46
Métro Raspail - Parking à la hauteur du 120 bd du Montparnasse
ouvert tous les jours lundi compris, sans interruption de 9 h 30 à 18 h 50, samedi fermeture à 17 h 50.

HAIFA : un amper-interpréteur complet

Olivier Herz

HAIFA (Herz Ampersand Interpreter For Applesoft) est, comme son nom l'indique, un utilitaire qui permet d'augmenter la capacité de l'interpréteur Applesoft de l'Apple II en utilisant pour cela l'Ampersand (&) qui permet d'accéder au langage machine grâce à un JMP en $3F5. HAIFA utilise aussi la fonction Applesoft USR (0) à laquelle on accède par un JMP en $A, JMP programmé par les soins d'une routine Ampersand.

Préoisons que HAIFA est spécialement destiné à être utilisé dans un programme plutôt qu'au clavier: il est donc inutile d'essayer d'y trouver des commandes utilitaires comme l'édition des lignes de programme, les remplacements de chaînes de caractères, la programmation des touches avec des mots-clefs de l'Applesoft ou encore les références croisées.

Notons toutefois qu'HAIFA peut être compatible avec certains utilitaires de mise au point de programmes comme le PLE (voir Pom's 1) ou le APA du DOS ToolKit. Vous trouverez dans l'annexe 2 toutes les informations utiles en ce qui concerne l'emplacement en mémoire de HAIFA.). Au sujet du chargement de HAIFA, prière de se reporter à l'annexe 1.

HAIFA peut rendre trois sortes de services aux personnes qui programment en Applesoft. Primo, faire tourner les programmes plus vite (routines graphiques principalement). Secundo, donner accès à des instructions qui font défaut en Applesoft (le GOTO NNN, le IF THEN ELSE, etc.). Tertio, simplifier et compacter l'écriture des programmes (& WAIT au lieu d'une boucle FOR NEXT pour une pause, tortue graphique, etc.).

Remarquons qu'HAIFA a été spécialement réalisé pour utiliser au mieux les capacités d'un Apple II avec Applesoft en ROM et carte langage; mais nous avons assemblé une version 48 K pour que tous puissent en profiter, et ce aux dépens de la place mémoire disponible et de la taille des tables utilisées par HAIFA.

HAIFA utilise au maximum les routines de l'Applesoft, mais ne touche pas au DOS qui, rappelons-le, réside en mémoire vive. En ce qui concerne les deux commandes DOS de HAIFA, l'une ne fait qu'un appel à RWTS ($3D9), et l'autre utilise des ordres DOS au niveau global, par l'équivalent en langage machine de PRINT D$.

Signalons enfin que les commandes de HAIFA utilisent un maximum de mots-clefs de l'Applesoft pour des raisons mnémotechniques et surtout parce que la commande LIST de l'Applesoft détache clairement ces mots-clefs en les entourant d'espaces.

LES COMMANDES GENERALES

La reprogrammation du RESET

Le RESET peut être reprogrammé si on lui fournit en $3F2 une adresse vers laquelle il doit sauter. Pour éviter de rebooter à chaque RESET, il faut aussi mettre en $3F4 l'EOR entre $3F3 et =$A5. Voir à ce sujet le manuel de référence de l'Apple II.

Avant ce saut de reprogrammation, le RESET effectue quelques commandes sous le contrôle du moniteur. Il met en mode TEXT et NORMAL, déconnecte les routines spéciales d'entrée/sortie, puis il agit comme CTRL-C : il fait un bip et indique un BREAK depuis le clavier ou un BREAK IN LINE XXX depuis un programme (il peut aussi dans ce cas être récupéré par une routine de ONERR GOTO). Ceci est réalisé par un JMP à l'adresse $D863, après avoir redonné le contrôle des entrées/sorties au DOS par un JMP $3EA.

Cette reprogrammation offre l'avantage de permettre à l'utilisateur de savoir à quel endroit il a arrêté le programme en appuyant sur RESET. C'est particulièrement utile quand on utilise HAIFA, étant donné qu'il peut y avoir des routines en langage machine assez longues, alors qu'une interruption par CTRL-C n'est testée qu'à chaque nouvelle instruction Applesoft.

L'initialisation

& INIT initialise HAIFA

Cette instruction est automatiquement réalisée quand on lance HAIFA. Mais il est conseillé de l'utiliser au début d'un programme, même si HAIFA est déjà chargé.

Cette instruction initialise les adresses de début des tables (musique, shapes, caractères haute résolution...) avec les valeurs

standard de HAIFA (voir l'annexe 2).

De plus, elle initialise la couleur haute résolution à 0 et le numéro de la page haute résolution utilisée à 1. (Cela ne sert à rien d'initialiser la couleur basse résolution car l'octet où est stocké cette couleur est aussi utilisé pour autre chose par les routines haute résolution de l'Applesoft.

Enfin, elle effectue les initialisations des routines & HGR et autres (voir plus bas) sans toutefois ni afficher ni remplir de page graphique haute résolution. Au contraire, elle met l'Apple II en mode TEXT, efface l'écran et met en place la routine normale d'entrée/sortie de caractères de HAIFA (voir plus bas).

La pile

Une grande innovation de HAIFA est la pile, qui permet d'accéder à un semblant de récursivité en autorisant la définition de variables locales.

& PUSH = [nom de variable] empile le descripteur de la variable.
& PULL = [nom de variable] dépile le descripteur de la variable.

Le descripteur occupe 5 octets de la façon suivante :

Réel	Entier	Chaîne
exposant:1	valeur:2	longueur:1
mantisse:4	inutilisés:3	adresse :2
		inutilisés:2

Si l'on essaye de sortir des limites de la pile, on obtient un OUT OF MEMORY ERROR.

On peut empiler une variable d'un type et en dépiler une d'un autre. Les résultats sont imprévisibles.

Attention, si l'on empile une chaîne. La chaîne peut ne plus être pointée par une variable : un nettoyage de mémoire (garbage collection) automatique ou forcé par le programme peut alors l'effacer: un & RECALL redonne alors n'importe quoi.

Les instructions de branchement

& AT [nom de variable entière] note dans le descripteur de la variable l'adresse mémoire du début de l'instruction suivante.

& TO [nom de variable entière] fait un saut à l'instruction pointée par le descripteur de la variable.

Si la variable n'est pas entière, on obtient un TYPE MISMATCH ERROR.

Malgré l'inconvénient de passer sur un & AT pour noter la valeur à donner au & TO, ce branchement possède trois avantages sur le GOTO : une grande rapidité, la possibilité de sauter au milieu d'une ligne et la possibilité de s'affranchir d'un numéro de ligne en ne tenant compte que de l'emplacement dans le programme de l'endroit à rejoindre.

Notons toutefois que le programmeur averti peut se passer du & AT s'il connait l'adresse mémoire de l'instruction à rejoindre. Il suffit alors d'affecter cette valeur à une variable entière.

Ces commandes permettent de simuler le REPEAT UNTIL du PASCAL ainsi que le DO WHILE à condition dans ce dernier cas de connaître l'adresse du début de la boucle et de commencer la boucle par un saut au test de fin:

```
REPEAT                  & AT I%
//////                  ///////
//////                  ///////
UNTIL [expression]      IF NOT [expression]
                        THEN & TO I%

WHILE [expression]      IF NOT [expression]
BEGIN                   THEN & TO I%
//////                  ///////
END;                    (position de I%)
```

& GOTO [expression numérique] est l'instruction de l'INTEGER BASIC.
& GOSUB [expression numérique] est l'instruction de l'INTEGER BASIC.

Si la ligne cherchée n'existe pas, on obtient un UNDEF'D STATEMENT ERROR.

& GOTO NEXT [expression numérique]
& GOSUB NEXT [expression numérique]

C'est pratiquement la même chose, à la différence près que, si la ligne cherchée n'existe pas, l'instruction renvoie à la première ligne ayant un numéro supérieur (s'il n'y en a pas, on obtient alors un UNDEF'D STATEMENT ERROR).

Le RESTORE NNN

& RESTORE [expression numérique] déplace le pointeur de DATA.

Le pointeur se positionne au début de la ligne cherchée. Si elle n'existe pas, on obtient un UNDEF'D STATEMENT ERROR. Il n'est pas nécessaire d'avoir des DATA sur la ligne pointée. A la première instructon READ, le programme recherchera la première instruction DATA suivant cette ligne. S'il n'en trouve pas, il affichera OUT OF DATA ERROR.

& RESTORE NEXT [expression numérique]

Mêmes remarques que pour le & GOTO NEXT.

Le listage généralisé

& LIST ou & LIST , listent tout le programme.
& LIST , [expression 2] liste du début du programme à l'expression 2.
& LIST [expression 1] , liste de l'expression 1 à la fin du programme.
& LIST [expression 1] , [expression 2] liste le programme entre les deux expressions.

Ces commandes ressemblent à celles de l'Applesoft. Toutefois, comme on a des expressions numériques, le signe "-" n'est plus autorisé à la place de la virgule. Lorsque l'expression 1 ne correspond à aucune ligne de programme, on obtient un UNDEF'D STATEMENT ERROR. Pour y remédier, il suffit de remplacer & LIST par & LIST NEXT. Voir à ce sujet les remarques du & GOTO NEXT. Lorsqu'il n'y à pas d'expression 1, le NEXT donne une SYNTAX ERROR.

Le "INPUT anything"

& INPUT " [chaîne de caractères] " ; [nom de variable]
& INPUT [nom de variable] permettent de rentrer une chaîne quelconque depuis le clavier.

Même syntaxe que le INPUT Applesoft à la différence près qu'on peut rentrer des guillemets, une virgule ou les deux points sans obtenir le fatidique message EXTRA IGNORED : on ne peut par conséquent rentrer qu'une seule variable par instruction.

Si la variable n'est pas une variable de chaîne, on obtient un TYPE MISMATCH ERROR. A partir du clavier, on obtient un ILLEGAL DIRECT ERROR.

Le GET numérique sans erreur

Un inconvénient fâcheux de l'Applesoft fait que si l'on presse une touche non numérique en réponse à un GET dont la variable est numérique, on obtient une SYNTAX ERROR.

& GET [nom de variable] permet d'affecter une valeur à la variable.

La variable doit être réelle ou entière sous peine de TYPE MISMATCH ERROR. L'instruction attend qu'une touche numérique soit appuyée. Une autre touche n'a aucun effet.

& GET + [nom de variable] affecte encore une valeur à la variable.

Ici, par contre, lorsqu'une touche non numérique est enfoncée, l'instruction assigne la valeur 0 à la variable.

Routine de ONERR GOTO

& ONERR exécute la routine fournie dans le manuel de référence Applesoft, à utiliser dans un sous-programme de traitement d'erreur.

& ONERR + effectue la même routine et envoie le message sonore d'erreur sans toutefois interrompre le programme.

Notons cependant quelques différences avec les messages classiques. Tout d'abord, comme HAIFA tient à se démarquer du DOS, les erreurs de ce dernier sont signalées par un DOS ERROR. Un enfoncement de CTRL-C est signalé par un BREAK ERROR et enfin une erreur dans un INPUT est signalée par un INPUT ERROR.

Détruire un tableau

& DEL [nom de tableau] détruit ce tableau.

Le tableau peut être réel, entier ou alphanumérique. S'il n'existe pas, on obtient un OUT OF DATA ERROR.

Changement d'une ligne de programme

& NEW [expression numérique] , [expression chaîne] remplace la ligne dont le numéro est l'expression numérique par l'expression chaîne.

Si cette instruction est exécutée à partir du clavier, on obtient une ILLEGAL DIRECT ERROR. S'il n'y a pas de numéro de ligne correspondant, on obtient le message UNDEF'D STATEMENT ERROR.

La ligne originale doit être plus longue que la chaîne, sous peine de STRING TOO LONG ERROR. Il est donc conseillé d'y mettre initialement un REM très long. Comme la ligne initiale est plus longue que la chaîne, HAIFA met à la fin de la chaîne ": REM" afin de pouvoir ignorer la fin de la ligne.

La chaîne ne doit bien sûr pas comprendre le numéro de ligne.

Cette instruction est un outil très puissant pour un programme d'initiation au BASIC ou pour un programme nécessitant d'entrer des formules mathématiques. Il est conseillé de l'utiliser avec l'ONERR GOTO actif.

ATTENTION aux fonctions et aux expressions chaînes définies par une chaîne de caractères dans la ligne initiale: il ne faut pas perdre de vue que les descripteurs de telles variables pointent directement vers une adresse du programme : changer la ligne modifie ces variables de manière imprévisible.

& NEW NEXT [expression numérique] , [expression string]

Mêmes remarques que pour le & GOTO NEXT.

Entrée des commandes moniteur

& CALL [chaîne] exécute les commandes moniteur que constitue la chaîne.

A partir du clavier, on obtient ILLEGAL DIRECT ERROR. HAIFA ajoute N D9C6C à la chaîne pour renvoyer au programme BASIC. Le délimiteur N fait que cette routine restitue le mode NORMAL.

Echange de deux variables

& SWAP ([nom de variable 1] , [nom de variable 2]) échange les descripteurs des deux variables.

Cette instuction est très rapide, car elle n'utilise pas comme en BASIC une variable intermédiaire. Si les deux variables ne sont pas du même type, on obtient un TYPE MISMATCH ERROR. Si l'on désire échanger les descripteurs de variables de types différents, il faut passer par la pile.

Le REPEAT PRINT

& REPT PRINT [chaîne] , [expression numérique] imprime la chaîne autant de fois qu'in- diqué par le résultat de l'expression numé- rique.

S'il y a un point-virgule à la fin de l'instruction, on n'imprime pas de retour chariot. L'expression numérique doit être comprise entre 0 et 255.

Le PRINT USING

& PRINT USR [chaîne] ; [expression numérique] , ... , [expression numérique] imprime les expressions numériques selon le format indiqué par la chaîne.

S'il y a un point-virgule à la fin de l'instruction, on n'imprime pas de retour chariot.

Caractéristiques du format, l'image selon laquelle le nombre résultant de chaque expression numérique sera imprimé :

. Le nombre sera placé dans le champ défini par le format, de droite à gauche, avec justification à droite.

. Le premier point à partir de la droite indique l'emplacement du point décimal et donc le nombre de décimales (notons que l'instruction arrondit).

. Les blancs, zéros, dollars ou étoiles indiquent les endroits où l'on peut placer des chiffres. Tous les autres caractères sont non-remplaçables.

. Les virgules seront imprimées dans le nombre, sauf celles à gauche du nombre qui seront remplacées par le caractère juste à leur droite (qui sera lui aussi imprimé selon les présentes règles).

Si un (ou plusieurs) chiffre du nombre est imprimé à la place d'un dollar du format, un dollar (unique) sera poussé à gauche par le nombre. D'où les possibilités de dollars fixes ou flottants (un dollar est fixe lorsqu'il n'y à pas de chiffre à sa position). Pour avoir des zéros ou des étoiles en tête du nombre, il suffit de remplir le champ du format avec ces caractères.

Le nombre imprimé peut enfin comporter jusqu'à neuf chiffres significatifs. Si le nombre est trop long (plus de neuf chiffres) ou si le champ est trop petit (pas assez de place pour le remplir avec les chiffres du nombre), le champ est entièrement rempli de dièses qui indiquent un dépassement de capacité. C'est préférable à une simple troncature, qui peut induire en erreur.

Faire une pause

& WAIT ([expression numérique]) permet de faire une pause.

L'expression doit être comprise entre 0 et 255 : la pause a pour durée la valeur de l'expression en dixiémes de seconde.

& WAIT + ([expression numérique]) fait la même chose à la différence près que la pression d'une touche permet de terminer prématurément la pause.

Notons que le clavier est remis à zéro à la fin de la routine.

PEEKs et POKEs généralisés

& POKE [expression 1] , [expression 2] permet de "poker" sur deux octets.

Les expressions doivent être comprises entre -65535 et 65535. La première représente bien sûr l'adresse et la deuxième un nombre entier de deux octets dont on met celui de poids faible dans l'octet de l'adresse et celui de poids fort dans l'octet suivant. Cette instruction permet par exemple de donner en une seule commande l'adresse de début de la table de SHAPES par un & POKE 232, adresse.

& PEEK programme le JMP de la fonction USR vers la routine de PEEK généralisé.

Il suffit alors de faire USR ([expression numérique]). L'expression, comprise entre -65535 et 65535, est comme ci-dessus une adresse et le résultat de la fonction USR est un entier de deux octets signé (donc compris entre -32768 et 32767) dont l'octet de poids faible est la valeur de l'octet de l'adresse et celui de poids fort de l'octet suivant.

Le nettoyage de la mémoire

& CLEAR permet de vider la mémoire des

chaînes de caractères inutilisées à condition que la quantité de mémoire vive encore disponible soit inférieure à 1 K.

& CLEAR + permet de le faire dans tous les cas.

Cette opération de nettoyage est la même que celle effectuée par l'Applesoft, soit forcée par l'utilisateur s'il invoque la fonction FRE(0), soit exécutée par le programme quand il sent la place mémoire lui manquer. Mais, au prix de l'emploi de 48 octets de stockage, cette routine est beaucoup plus rapide que celle de l'Applesoft, car au lieu de travailler chaîne de caractères par chaîne de caractères, elle en traite 16 à la fois, selon le principe expliqué dans Pom's, numéro 2.

Le IF THEN ELSE

& IF [expression] THEN [instructions] : & ELSE : [instructions] est l'instruction bien connue des utilisateurs de BASIC Microsoft.

Notons d'abord qu'à la différence du IF de l'Applesoft, celui-ci ne tolère pas une chaîne car cela ne sert à rien. Remarquons bien que le & IF et le & ELSE sont sur la même ligne.

Lorsque l'expression est vraie, c'est-à-dire différente de 0, toutes les instructions entre le & IF et le & ELSE sont effectuées et celles après le & ELSE ignorées. Lorsqu'elle est fausse, c'est-à-dire lorsqu'elle vaut 0, seules les instructions après le & ELSE sont effectuées. Enfin, à moins d'un renvoi par GOTO, RETURN, & TO etc., les instructions de la ligne suivante sont effectuées ensuite dans les deux cas. Lorsqu'il n'y à pas d'instrutions après le & ELSE ou lorsque cette dernière instruction n'accompagne pas le & IF, on à l'équivalent d'un IF classique.

Remarquons qu'on ne peut emboîter plusieurs IF THEN ELSE car HAIFA recherche le premier & ELSE aprèes le & IF. Mais on peut cependant y emboîter des boucles IF dans la limite de la capacité de l'interpré- teur, c'est-à-dire une ou deux. Signalons aussi qu'un & ELSE qui ne suit pas un & IF sur une ligne est considéré par HAIFA comme un REM. On a ainsi la possibilté d'avoir des REM d'instructions BASIC bien mises en place au listing et qui sont, comble de chance, changées par RENUMBER quand il s'agit de GOTO ou de GOSUB.

Enfin, le THEN a été rendu faculatif par HAIFA car il faut bien reconnaître qu'il ne sert à rien. Lorsqu'il y a un THEN suivi par un numéro de ligne, il signifie encore GOTO.

Choix à partir d'un menu

& LET ([expression 1] , [expression 2]) = [nom de variable] effectue la routine de menu.

La variable doit être entière ou réelle: à la fin de la routine, elle contiendra le numéro de l'item choisi.

Les expressions, qui sont numériques, représentent respectivement le nombre de lignes à sauter entre chaque item (par exemple, si c'est 1, il y aura un item toutes les deux lignes) et le nombre d'items. Le premier item est situé sur la première ligne de la fenêtre de la page texte et le dernier doit se trouver encore dans cette fenêtre, sous peine de ILLEGAL QUANTITY ERROR. Les items sont numérotés de 0 à la valeur de l'expression 2.

La routine affiche alors une barre en inverse sur le premier item, sur toute la longueur de la fenêtre texte. En fait, les caractères en mode NORMAL sont repreésentés en INVERSE et vice-versa, alors que les caractères FLASH et minuscules restent inchangés. L'utilisateur peut alors 'promener' la barre sur les différents items en utilisant la flèche à droite pour descendre et celle à gauche pour monter. Il y a d'ailleurs 'bouclage' quand on arrive en haut ou en bas. Appuyer sur ESCAPE permet de sortir de cette routine en affectant 0 à la variable: aucun item n'a été choisi. Appuyer sur RETURN sort également, mais affecte à la variable le numéro de l'item choisi (celui qui était sous la barre). Pour choisir l'item, on peut aussi, si son numéro est inférieur ou égal à 9, appuyer sur le chiffre correspondant.

Il est de la responsabilité du programmeur d'afficher les noms et éventuellement le numéros des items avant de lancer le menu, ainsi que d'adapter la fenêtre de la page texte à ses besoins pour effectuer cette routine de menu. Signalons enfin que, quelle que soit la manière dont on quitte la routine (sauf RESET bien entendu!), l'écran est rendu à son état initial, car appliquer deux fois la barre laisse le statu quo. D'autre part, cette routine est incompatible avec la sortie de caractères sur la page haute résolution car elle POKE directement la barre sur la page texte.

& LET ([expression 1] , [expression 2] , [expression 3]) = [nom de variable] fait la même chose mais remplace la barre par l'impression dans la première colonne de la fenêtre du caractère dont le code écran correspond à l'expression 3 (de 0 à 255).

Le caractère de l'expression 3 peut par exemple être une flèche à droite en FLASH, de code écran 126 (voir le manuel de référence de l'Apple II pour connaître les codes écran). Comme la routine POKE le caractère sur la page texte, il y a toujours incompa-

tibilité avec la sortie de caractères sur la page haute résolution. Notons également que l'écran est laissé intact à la fin, la routine retenant en mémoire le caractère se trouvant là où elle veut mettre le caractère de l'expression 3.

2. LES COMMANDES GRAPHIQUES BASSE RESOLUTION

Les initialisations

& GR ([expression numérique]) initialise en mode mixte.
& GR + ([expression numérique]) initialise en mode non-mixte (avec les quatre lignes du bas).

L'expression numérique identifie la couleur, comprise entre 0 et 255; mais seule la valeur modulo 16 est prise en considération. L'écran est rempli avec cette couleur.

& GR et & GR + font la même chose, mais sans remplir l'écran et sans changer la couleur utilisée.

Les shapes

Le format de la table de shapes est le même qu'en haute résolution : la routine d'analyse syntaxique est d'ailleurs calquée sur celle utilisée en haute résolution.

1. Index

```
( début = S+0   n (0-$FF)   ) nombre de shapes
(         S+1   inutilisé
(         S+2   octet bas   ) adresse du pre-
          mier octet de la shape 1,
(         S+3   octet haut  ) relatif à l'a-
          dresse S
(               ...........
(        S+2n   octet bas   ) adresse du pre-
          mier octet de la shape n,
(      S+2n+1   octet haut  ) relatif à l'a-
          dresse S
```

2. Octets des shapes

```
(         S+D1   1er octet   )
(                .........   ) shape 1
(       S+D2-1   dernier=0   )
(                .........
(         S+Dn   1er octet   )
(                .........   ) shape n
(               dernier=0   )
```

De même qu'en haute résolution, on plotte avant d'effectuer le déplacement, lors de la description de la shape. Mais, cela mis à part, la description de la shape est différente.

La partie basse de l'octet (de 0 à F) contient la couleur du point. Elle est inutilisée si l'on ne plotte pas. On a préféré faire un léger gâchis de place mémoire dans la table de shapes, plutôt que de rallonger et compliquer la routine. La partie haute de l'octet (de 0 à F) indique le déplacement:

Sans plotter	En plottant
7 0 1	F 8 9
6 * 2	E * A
5 4 3	D C B

& DRAW [expression 1] AT [expression 2] , [expression 3] dessine la shape avec la couleur basse résolution en vigueur.
& XDRAW [expression 1] AT [expression 2] , [expression 3] dessine la shape avec les couleurs indiquees dans la table de shapes.

Les expressions sont numériques. La première, de 0 à n, représente le numéro de la forme (shape). La deuxième, comprise entre 0 et 39, et la troisième, comprise entre 0 et 47, représentent le point de départ du tracé de la forme. Lorsqu'on dépasse les limites de l'écran, les points en question ne sont pas plottés. Le point de départ doit toutefois être sur l'écran.

Lorsqu'on est en mode mixte, une forme débordant dans la zone texte y est quand même plottée, engendrant ainsi des caractères écran divers. Il vaut donc mieux faire un HOME dans ce cas.

Il n'y a pas de shape 0, bien que l'expression 1 le tolère. C'est le même bug qu'en haute résolution, puisque les routines sont semblables.

3. GENERALITES HAUTE RESOLUTION

Les initialisations

& HGR ([expression numérique]) initialise la page 1 en mode mixte.
& HGR + ([expression numérique]) initialise la page 1 en mode non-mixte (avec les quatre lignes du bas).
& HGR2 ([expression numérique]) initialise la page 2 en mode non-mixte.

Le mode mixte ne sert pas pour la page 2 car la page 2 de texte n'est généralement pas utilisée comme telle. L'expression doit être comprise entre 0 et 7. Elle identifie la couleur haute résolution qui remplit la page considerée.

Les initialisations mettent la tortue en 0,0 avec une rotation nulle, font ROT = 0 et SCALE = 1, initialisent la fenêtre à la taille de l'écran, mettent le mode égal à 255,0,0, mettent le type égal à 10, mettent la position d'impression de caractères haute

résolution en 0,0 et suppriment l'éventuelle sortie de caractères haute résolution en installant le mode normal d'entrée/sortie de caractères au clavier et à l'écran de HAIFA. Voir plus bas la signification de ces expressions.

& HGR , & HGR + et & HGR2 font la même chose, mais sans remplir l'écran ni changer la couleur utilisée.

Le mode de remplissage

& MODE = [expression 1] , [expression 2] , [expression 3]

Les expressions sont numériques, de 0 à 255. Certaines routines haute résolution recopient un octet de l'écran à sa place ou à un autre endroit. Auparavant, HAIFA fait un AND logique entre cet octet et l'expres- sion 1, un OR entre le résultat et l'ex- pression 2 puis un EOR avec l'expression 3. Les valeurs par défaut du mode sont celles qui n'ont aucun effet: 255 pour l'AND, 0 pour l'OR et 0 pour l'EOR. Par exemple 127,0,0 met à 0 les 8èmes bits de couleur; 255, 128,0 les met à 1; et 255, 0, 128 les inverse. Ou encore 255, 0, 127 inverse la fenêtre haute résolution, les 8èmes bits étant inchangés et 255,0,255 inverse la fenêtre, 8èmes bits compris.

Le type de remplissage

& TYPE = [expression numérique]

L'expression est comprise entre 0 et 255. Les routines de sortie de caractères haute résolution envoient un octet d'une table sur un octet de l'écran après avoir fait une opération logique entre ces deux octets. Si le type est supérieur ou égal à 128, on met à 1 les 8èmes bits de couleur de la table avant l'opération; sinon, on les met à 0. Et le type MODULO 16 est alors le même qu'en PASCAL:

Type	Effet
0	0
1	T NOR E
2	T AND NOT E
3	NOT E
4	NOT T AND E
5	NOT T
6	T EOR E
7	T NAND E
8	T AND E
9	T [=] E
10	T
11	T OR NOT E
12	E
13	NOT T OR E
14	T OR E
15	1

où T désigne un octet de la table et E celui de l'écran se trouvant à sa destination. La valeur par défaut est celle qui copie simplement les octets de la table à l'écran (10).

Les options haute résolution

Diverses options peuvent être mises dans certaines instructions haute résolution. Lorsqu'il y en a plusieurs, elles doivent se suivre dans l'ordre correspondant à leur présentation ci-dessous.

L'OPTION PRIME: mettre un prime (') juste après la commande fait que l'on change la page graphique principale, sans changer toutefois la page affichée. Tout ce qui devait sortir sur la page 1 sort sur la page 2 et vice-versa. Cette option est en quelque sorte un POKE 230,96 - PEEK (230) amélioré. Elle permet de faire du 'ping-pong' en écrivant sur la page non visible qui sera affichée, une fois mise au point, ce qui évite le clignotement.

L'OPTION POUR-CENT (%) est utilisée dans les routines qui se servent du MODE. Les AND, OR et EOR sont appliqués aux octets de la page pricipale (éventuellement modifiée avec l'option prime), et le résultat est recopié dans l'autre page.

L'OPTION DIESE (=) est utilisée dans les routines qui tracent des lignes ou dans la routine de sortie de caractères & DRAW PRINT. Elle donne l'ordre à HAIFA de ne pas dessiner les points qui se trouvent en dehors de la fenêtre haute résolution. C'est ce qu'on appelle le 'clipping'. Cela n'empê- che pas dans le cas du tracé de lignes d'avoir un ILLEGAL QUANTITY ERROR si l'on essaye de dessiner hors de l'écran, que la fenêtre coïncide ou non avec l'écran.

La fonction SCRN haute résolution

Les points sont rangés sur l'écran par octets de sept points horizontaux (les sept bits les plus bas de l'octet, le bit le moins significatif représentant le point le plus à gauche). Le huitième bit de l'octet (de poids fort) représente la couleur):

	(noir 1	(couleur 0)
bit	(jaune	(couleur 1)
=0	(bleu	(couleur 2)
	(blanc 1	(couleur 3)
	(noir 2	(couleur 4)
bit	(vert	(couleur 5)
=1	(mauve	(couleur 6)
	(blanc 2	(couleur 7)

Les couleurs indiquées ci-dessus sont celles de la carte RVB classique. Pour les utilisateurs du 'chat mauve', regarder le mode d'emploi de cette carte.

Lorsqu'un point est allumé, le bit le

représentant est à 1. Les points des colonnes 2n sont bleus ou mauves, ceux des colonnes 2n+1 jaunes ou verts. Lorsque sur une même ligne le point de la colonne 2n et celui de la colonne 2n+1 sont allumés, l'ensemble parait blanc. Lorsqu'aucun n'est allumé, l'ensemble est noir.

& SCRN ([expression 1] , [expression 2]) = [nom de variable] assigne à la variable la valeur décrite ci-dessous.

Les expressions sont numériques et la variable doit être réelle ou entière.

point d'une colonne 2n			
SCRN	état du 8e bit	état du point	état du voisin
0	0	0	0
1	0	0	1
130	0	1	0
131	0	1	1
4	1	0	0
5	1	0	1
134	1	1	0
135	1	1	1

point d'une colonne 2n+1			
SCRN	état du 8e bit	état du point	état du voisin
0	0	0	0
129	0	1	0
2	0	0	1
131	0	1	1
4	1	0	0
133	1	1	0
6	1	0	1
135	1	1	1

Ainsi le point est allumé si et seulement si le résultat de & SCRN est supérieur à 128. Et la partie modulo 128 donne la couleur (de 0 à 7) visible à cet endroit.

4. LA FENETRE HAUTE RESOLUTION

Deux choix s'offraient: soit définir une fenêtre haute résolution au point près (de 0 à 279 horizontalement et de 0 à 191 verticalement), soit utiliser l'équivalent des positions de la page texte (de 0 à 39 horizontalement et de 0 à 23 verticalement) avec des positions-types horizontales multiples de 7 (rappelons qu'un octet horizontal contient 7 points) et verticales multiples de 8 (voir pourquoi dans le manuel de référence Apple II).

La première possibilité offrait l'avantage d'être plus précise et d'éviter certaines 'bavures' dues à la bizarre représentation des couleurs, mais elle avait l'inconvénient d'ignorer pratiquement le 8ème bit de couleur, car ce bit concerne sept points à la fois.

La deuxième possibilité, moins précise, permet au contraire de se servir de ce bit et possède aussi l'avantage de correspondre à des routines beaucoup plus courtes et beaucoup plus simples à réaliser. C'est donc cette solution qui à été retenue.

Dimensionner la fenêtre

& DIM = [expression 1] , ... , [expression 4] donne la dimension de la fenêtre.

Les quatre expressions sont numériques, respectivement la gauche (de 0 à 39), la droite (de 0 à 39), le haut (de 0 à 23) et le bas (de 0 à 23) de la fenêtre. Les limites sont incluses dans la fenêtre.
On doit avoir la valeur de l'expression 1 (gauche) inférieure ou égale à celle de l'expression 2 (droite) sous peine de ILLEGAL QUANTITY ERROR. De même pour les valeurs des expressions 3 (haut) et 4 (bas).

Remplir la fenêtre

& FILL ([expression 1] , ... , [expression 4]) remplit la fenêtre point par point. L'option prime est permise.

Les quatre expressions sont numériques, de 0 à 7: ce sont les couleurs avec les- quelles on remplit la fenêtre. On remplit les lignes alternativement avec les couleurs 1 et 2, puis 3 et 4. Et sur chaque ligne on remplit les colonnes 2n et 2n+1 de façon à ce que l'on voie la bonne couleur (1 et 2, ou 3 et 4 alternativement). Les couleurs 1 et 2, 3 et 4, puisqu'utilisées dans un même octet, doivent avoir le même bit haut. Si ce n'est pas le cas, c'est le bit haut de 2 ou de 4 qui est pris en considération. Exemples avec deux couleurs a et b: & FILL (a,a,b,b) remplit en alternant les couleurs ligne par ligne; & FILL (a,b,a,b) remplit en alternant les couleurs colonne par colonne; & FILL (a,b,b,a) remplit en quinquonce.

ATTENTION AUX EFFETS DE BORD: comme les octets contiennent un nombre impair de points et qu'un point visible utilise deux points écran (colonnes 2n et 2n+1), il peut y avoir des effets de bord aux limites gauche ou droite de la fenêtre. Pour les éviter, le bord gauche de la fenêtre doit être pair et le bord droit impair. Ceci ne concerne que la carte RVB classique. Il n'y a pas de probleme en noir et blanc ou avec 'le chat mauve'. Toutefois, avec cette der- nière carte, un remplissage en quinquonce parait noir et blanc.

& SQR ([expression 1] , ... , [expression 4]) remplit la fenêtre carré par carré. L'option prime est permise.

Les quatre expressions sont numériques, de 0 à 7: ce sont les couleurs avec les- quelles on remplit la fenêtre. On remplit les lignes alternativement avec les couleurs 1 et 2, puis 3 et 4; et les colonnes avec 1 et 3, puis 2 et 4. A la différence du & FILL, les lignes et les colonnes correspondent à la page texte. Au lieu de remplir point par point, on remplit carré par carré, où les carrés sont les carrés élémentaires 7 x 8.

ATTENTION AUX EFFETS DE BORDS: hormis les mêmes effets de bord que le & FILL (voir ci-dessus), il y a des effets de bord à l'intérieur de la fenêtre entre les carrés horizontaux (même avec la carte 'le chat mauve'). Les couples qui les évitent sont, hormis ceux indiquant la même couleur au bit haut près: (0-4, 1-5, 2-6 et 3-7), 0-1, 0-2, 0-5, 3-1, 3-2, 3-6, 4-1, 4-5, 4-6, 7-2, 7-5 et 7-6.

Bouger et modifier la fenêtre

& COPY recopie la fenêtre à sa place en utilisant le mode. Les options prime et pour-cent sont permises.

& SHFT [caractère] décale la fenêtre d'une position.
& CLR [caractère] décale d'une position l'intérieur de la fenêtre, en la vidant.
& CRCL [caractère] fait une permutation circulaire de l'intérieur de la fenêtre.

Les options prime et pour-cent sont permises et le mode est utilisé pour le remplissage. Si le décalage de & SHFT est empêché par la présence d'un bord de l'écran, on obtient une ILLEGAL QUANTITY ERROR. Le caractère indique la direction: " " pour la gauche, " " pour la droite, "+" pour le haut, "-" pour le bas. Un autre caractère donne ILLEGAL QUANTITY ERROR. Lorsque l'option pour-cent n'est pas mise, les trous laissés par & SHFT ou & CLR sont remplis par la couleur haute résolution en usage.

Comme les octets contiennent un nombre impair (7) de points horizontaux, un déplacement horizontal modifie les couleurs (échange de 1 et 2, de 5 et 6). Pour éviter cela, il est conseillé de faire deux déplacements consécutifs.

Les déplacements verticaux utilisent les 80 derniers octets du buffer d'entrée pour stocker des lignes graphiques: il faut donc éviter d'entrer depuis le clavier des lignes de plus de 176 caractères comportant des déplacements verticaux, sous peine de SYNTAX ERROR.

Les instructions & SHFT et & CLR modifient les dimensions de la fenêtre: (& SHFT fait une translation d'une unité et & CLR diminue la taille d'une unité, sauf dans le cas d'un & CLR quand la largeur de la fenêtre dans la direction considerée vaut 1 (dans ce cas elle reste à 1, au lieu de passer à 0).

5. LA TORTUE HAUTE RESOLUTION

HAIFA possède comme PASCAL une tortue haute résolution: aux coordonnées x et y du point courant de l'écran on ajoute la rotation a comprise entre 0 et 255 (pseudo-degrés):

```
        064
         !
         !
    128--*--000
         !
         !
        192
```

Fixer la rotation

& ROT = [expression numérique] fournit la valeur de a.
& ROT ([expression numérique]) ajoute l'expression à la valeur de a.

L'expression doit être comprise entre -65535 et 65535 et est traitée modulo 256.

Tracer des points et des droites

& (HPLOT) ([expression 1] , [expression 2])
& (X HPLOT) (TO [expression 1] , [expression 2])
& (N HPLOT) (ON [expression 3])

Les options prime et dièse sont permises.
Les expressions sont numériques, les deux premières représentant un point de l'écran (x de 0 à 279 et y de 0 à 191) et la troisième la distance que la tortue doit parcourir dans la direction a. On obtient une ILLEGAL QUANTITY ERROR si cela doit amener la tortue à sortir de l'écran.

& HPLOT est le HPLOT classique, à la seule différence que si l'on change la couleur avant un HPLOT TO, le HPLOT TO classique continue avec l'ancienne couleur, mais pas le & HPLOT TO.

& X HPLOT fait la même chose, mais en couleur inverse. Le bit haut reste inchangé; on a donc les échanges de couleurs 0-3, 1-2, 4-7 et 5-6. ATTENTION! Si faire & X HPLOT x1,y1 TO x2,y2: & X HPLOT x1,y1 TO x2,y2 laisse l'écran inchangé, il n'en est pas de même pour & X HPLOT x1,y1 TO x2,y2: & HPLOT x2,y2 TO x1,y1.

& N HPLOT fait la même chose mais sans laisser de trace sur l'écran (couleur nulle).

Les instructions multiples sont permises

avec TO et ON, par exemple: & HPLOT x1,y1 TO x2,y2 ON x3,y3 ON x4,y4 TO x5,y5. Notons que les valeurs par défaut aprés initialisation sont: x = y = a = 0.

6. LES ENTREES SORTIES

Les entrées/sorties disque

& FRE , S [chiffre] , D [chiffre] reprogramme la fonction USR pour la routine de secteurs libres.

Les commandes S et D, qui indiquent le slot et le drive, sont facultatives; leur ordre importe peu et il peut même y en avoir plusieurs, seule la dernière étant prise en compte. Si l'une des deux (ou les deux) n'y est pas, la valeur par défaut est celle du précédent & FRE s'il y en a eu un, ou 6 pour le slot et 1 pour le drive s'il n'y en n'a pas eu.

USR (0) fournit alors le nombre de secteurs libres dans la disquette correspondant au drive et au slot décrits ci-dessus.

& STORE [nom de tableau] , [chaîne] correspond à l'instruction STORE pour cassettes.

& RECALL [nom de tableau] , [chaîne] correspond à l'instruction RECALL pour cassettes.

Ces instructions utilisent des fichiers binaires sur disque.
Le tableau doit être numérique; s'il n'existe pas, on obtient une OUT OF DATA ERROR. La chaîne représente le nom du fichier qui stockera le tableau. Ce doit être un nom valide sinon le DOS émettra un message d'erreur.

Dans le cas d'un & RECALL, le tableau récepteur doit avoir les mêmes dimensions que celui qui avait été enregistré sur le fichier. Sinon, si le fichier est trop petit, la routine imprime ERR et fait un bip, mais n'arrête pas le programme; et s'il est trop gros au contraire, il y a des catastrophes car le fichier BLOADé déborde du tableau et la routine prend l'initiative de faire un CLEAR et de donner une OUT OF MEMORY ERROR. Notons que si le tableau récepteur a la même taille, mais avec des dimensions différentes (par exemple le tableau enregistré était 10x10 et le tableau récepteur est 20x5), il n'y a pas d'erreur, mais la situation risque de ne pas être très claire.
Quand ces instructions sont utilisées depuis le clavier, d'autres instructions qui auraient été mises sur la même ligne ne sont pas effectuées car le DOS efface le buffer d'entrée et ces routines retournent, quand on les appelle du clavier, au niveau général de l'Applesoft.

Ces instructions permettent l'enregistrement d'un tableau numérique sur disquette environ 16 fois plus rapidement qu'avec un fichier TEXT. D'autre part, la place nécessaire pour stocker le tableau sur disque a été réduite de plus de 50%.

Les entrées/sorties imprimante

& DUMP TEXT , S [chiffre] imprime la page texte sur l'imprimante.
& DUMP GR , S [chiffre] imprime la page graphique basse résolution sur l'imprimante.

Il s'agit dans les deux cas de la page 1, la page 2 étant rarement utilisée car elle correspond au début des programmes Applesoft. Le chiffre indique le slot. La commande S est facultative, la valeur par défaut étant celle du précédent & DUMP (l'un ou l'autre) effectué ou 1 s'il n'y en n'a pas eu. Cette instruction concerne exclusivement les imprimantes à interface parallèle qui constituent la majorité des imprimantes bas de gamme destinées à l'Apple II (Centronics, Epson, Seikosha, etc.).

Il est inutile d'avoir branché l'imprimante, le programme lui envoyant directement les caractères sans passer par les routines d'entrée/sortie du moniteur. Il est toutefois conseillé de l'avoir allumée!

Pour le & DUMP TEXT, les caractères FLASH ou INVERSE ne sont pas distingués par l'imprimante car l'instruction les rétablit en mode NORMAL avant l'envoi du code à l'imprimante.

Pour le & DUMP GR, si la couleur vaut 0 (c'est-à-dire le noir), HAIFA envoie un espace à l'imprimante et dans le cas contraire (couleur 1 à 15), il envoie le chiffre hexadécimal correspondant (1 à 9 puis A à F pour 10 à 15). Comme il y a deux fois plus de lignes que dans la page texte et que les caractères de l'imprimante n'ont q'une hauteur possible, il peut être conseillé, pour respecter approximativement les proportions de la page graphique, d'utiliser le mode double largeur de l'imprimante s'il existe. Pour cela, il faut ou bien faire un PR#1 et envoyer à l'imprimante par un PRINT le caractère de contrôle du mode double largeur ou bien POKEr l'adresse d'entrée/sortie de l'imprimante avec ce caractère ou bien encore envoyer à la routine de l'interface parallèle (celle-la même utilisée par HAIFA) ce même caractère de contrôle.

La musique

Toutes les routines musicales utilisent l'option prime. Si l'on met un prime (') juste après le nom de la commande, et avant ses arguments, le son sortira sur la sortie cassette. Si le prime est absent, il sort sur le haut-parleur interne de l'ordinateur.

& NOT ([expression 1] , [expression 2]) fournit la routine classique de musique.

Les expressions sont numériques, de 0 à 255. La première représente la hauteur de la note, 255 étant la plus grave. Cette expression est inverse de la fréquence en ce sens que doubler l'expression descend la note d'un octave.
La deuxièeme expression est proportionnelle à la durée de la note.

& POP ([expression 1] , [expression 2]) est une routine qui peut servir dans les jeux car elle donne des bruits de sirènes et de phasers.

Les expressions sont encore numériques de 0 à 255. La première représente en quelque sorte la longueur d'un coup (buzzer pour moins de 128 et phaser au-delà), et la seconde le nombre de coups. Notons que les coups ont une longueur qui augmente avec l'expression 1. Le mieux est que lecteur s'amuse lui-même à voir comment marche la routine.

& PLAY1 joue un air à une voix.

La table qui contient l'air est formée de groupes successifs de 3 octets qui représentent les notes. La hauteur de la note est représentée par le complément à 512 de la somme des deux premiers: encore une fois, doubler ce complément correspond à descendre d'un octave.

Le rapport de ces deux octets donne le timbre: classique si le rapport vaut 1 et de plus en plus nasillard au fur et à mesure que l'on s'éloigne de la valeur 1, dans un sens ou dans un autre.

Le troisième octet complémenté à 256 est proportionnel à la durée de la note. Ainsi, si les octets valent 200, 92 et 220, on a un LA (292) avec un timbre 2 et c'est une croche (36=256-220). Ces notes peuvent être prises pour référence, les autres se calculant d'après celles-là en connaissant les fréquences et les durées relatives.

DO	RE	MI	FA	SOL	LA	SI	DO
1	9/8	5/4	4/3	3/2	5/3	15/8	2

Si le premier octet vaut 0, le programme effectue une pause dont la longueur correspond au troisième octet. La nullité du troisième octet quant à elle indique la fin de l'air joué. Notons que l'utilisateur peut à tout moment interrompre la musique en appuyant sur une touche.

& PLAY2 joue un air à deux voix.

Il s'agit là de la version diffusée de la routine du programme ELECTRIC DUET qui n'a pu être introduite dans la version de HAIFA diffusée par POM's pour des raisons de copyright. Il s'agit encore de la lecture d'une table de groupes de 3 octets dont les deux premiers représentent les hauteurs des voix et le troisième la durée commune. Pour plus de détails, voir le mode d'emploi de ELECTRIC DUET. Cette routine peut être incorporée à HAIFA par tout possesseur de ELECTRIC DUET.

& PLAY1 ([expression numérique]) et
& PLAY2 ([expression numérique]) font la même chose que précédemment mais en commençant par la note dont le rang correspond à l'expression numérique.

Les entrées/sorties de caractères

HAIFA a reprogrammé les entrées/sorties de caractères à l'écran pour deux raisons: utiliser des caractères de contrôle pour faire certaines opérations et avoir facilement les minuscules au clavier et à la sortie écran. Rappelons cependant que les minuscules n'apparaissent comme telles à l'écran que pour les possesseurs d'une carte ou d'une ROM spéciale (voir le manuel de référence Apple II pour l'aspect des minuscules sans carte spéciale).

Pour quitter un mode d'entrée/sortie, il suffit de faire un PR= pour la sortie de caractères et un IN= pour l'entrée. Par exemple PR=0 et IN=0 redonnent le contrôle des entrées/sorties aux routines moniteur classiques (COUT1 pour la sortie et KEYIN pour l'entrée). On peut aussi, depuis le clavier, appuyer sur la touche RESET. Rappelons toutefois que HAIFA est essentiellement destiné à être utilisé à partir d'un programme.

Ces modes d'entrées/sorties sont repérés facilement par l'utilisateur car le curseur ne clignote plus et donc représente les caractères en INVERSE au lieu de les représenter en FLASH. Cela permet d'établir une similitude entre les entrées/sorties sur la page texte et sur la page graphique haute résolution; car dans ce dernier cas il n'est pas possible de générer un curseur clignotant sans ralentir considérablement la vitesse des routines d'entrées/sorties, puisqu'il faudrait réaliser ce clignotement par SOFT et non par HARD comme pour la page texte.

Dans tout ce qui suit, on désigne par caractère de contrôle tout caractère dont le code ASCII est compris entre 0 et 32, on parlera de lettre majuscule pour un caractère de code se situant entre 64 et 95 et de lettre minuscule entre 96 et 127.
Nous utiliserons aussi les notations suivantes pour les caractères suivants: crochet ouvrant = ASCII 91, backslash = 92,

crochet fermant = 93, underscore = 95 et SHIFT-P = 64.

Voyons un peu l'utilisation du clavier faite par HAIFA:

```
1 2 3 4 5 6 7 8 9 0 : - RESET
ESC Q W E R T Y U I O P REPT RETURN
CTRL A S D F G H J K L ; -- --
SHIFT Z X C V B N M , . / SHIFT
BARRE D'ESPACEMENT
```

Les entrées

Elles font une utilisation intensive des caractères de contrôle. Les caractères de contrôle suivants sont ignorés par la routine d'entrée en ce sens qu'elle reboucle sur elle-même si elle les rencontre et qu'elle ne les rentre pas dans le buffer d'entrée: CTRL-%, CTRL-crochet fermant et CTRL-SHIFT-P. Notons que CTRL-backslash et CTRL-underscore sont inaccessibles au clavier.

Les caractères de contrôle suivants sont considérés commes des caractères normaux par la routine d'entrée: CTRL-C qui permet d'interrompre un INPUT ou un & INPUT, CTRL-D qui invoque le DOS lorsqu'il est employé dans une instruction PRINT, CTRL-G qui émet un bip quand il est envoyé à une routine de sortie, CTRL-H qui est la flèche à gauche, CTRL-J qui fait un saut de ligne lorsqu'il est envoyé à la routine de sortie de caractères, CTRL-M qui est la touche RETURN, CTRL-U qui est la flèche à droite, CTRL-X qui permet d'annuler l'entrée dans les routines d'entrée d'une ligne de caractères.

Les caractères de contrôle suivants effectuent une action particulière mais sont ensuite ignorés par la routine d'entrée qui reboucle sur elle-même et ne les rentre pas dans le buffer d'entrée:

CTRL-A déplace le curseur à gauche comme ESC B, CTRL-S le déplace à droite comme ESC A, CTRL-W le déplace vers le haut comme ESC D et CTRL-Z vers le bas comme ESC C.

CTRL-P efface l'écran comme ESC SHIFT-P, CTRL-E efface jusqu'à la fin de la ligne comme ESC E et CTRL-F efface jusqu'au bas de l'écran comme ESC F. Notons que dans le cas d'une sortie sur l'écran haute résolution, ces caractères agissent de la même facon.

CTRL-R met en place le mode majuscule (qui est mis par défaut au lancement des routines d'entrées/sorties), CTRL-V met en place le mode minuscule et ESC fait que la prochaine lettre tapée sera majuscule, les suivantes restant minuscules (et donc les fonctions classiques de ESC sont déconnectées).

Les caractères suivants reprogramment la flèche à droite: CTRL-N fait qu'elle relira les lettres telles qu'elles sont (c'est le mode mis par défaut), CTRL-T fait qu'elle les relira comme étant des majuscules, CTRL-B fait qu'elle les relira comme étant des minuscules et CTRL-Y fait qu'elle relira les majuscules comme étant des minuscules et vice-versa.

Notons au passage que les commandes Applesoft et DOS doivent obligatoirement être rentrées en majuscules, sous peine de SYNTAX ERROR.

Enfin CTRL-@ fait que toute touche enfoncée émettra une courte note de musique d'autant plus grave que son code ASCII est élevé. Réappuyer CTRL-@ refera silence (mode par défaut).

Les caractères suivants permettent d'entrer des caractères inaccessibles au clavier; les caractères de contrôle sont alors transformés en ces caractères par la routine d'entrée: CTRL-K fournit le crochet ouvrant (ASCII 91 en majuscule et 123 en minuscule), CTRL-L fournit le backslash (ASCII 92 et 124) et CTRL-O fournit le underscore (ASCII 95 et 127).

CTRL-I permet d'insérer un caractère dans le buffer d'entrée, n'importe quel caractère pourvu qu'il soit accessible au clavier. A cette fin, le curseur disparait jusqu'à ce que l'on ait enfoncé une touche, touche qui ne sera d'ailleurs envoyée ni à la routine d'entrée ni à celle de sortie: cela permet par exemple d'insérer dans des chaînes de caractères ou dans des REM des carriage return, des flèches à gauche, les caractères spéciaux des routines entrées/sorties de HAIFA, etc.

Terminons par quelques remarques à propos des maniéres de rentrer des caractères: il y à d'abord l'entrée de lignes par la routine moniteur GETLN (utilisée par le niveau général du moniteur et de l'Applesoft ainsi que par les instructions INPUT et & INPUT) qui appellent à chaque fois la routine moniteur RDKEY, qui appelle alors à son tour la routine d'entrée de HAIFA ou autre; et il y a aussi les instructions GET et & GET qui se contentent d'un appel de RDKEY. Il faut donc bien voir que ces deux dernières instructions attendent un caractère qui "passe à travers" la routine d'entrée de HAIFA lorsque celle-ci est utilisée. En particulier, le CTRL-T n'est dans ce cas d'aucune utilité et la touche qu'il demande d'enfoncer ne sera pas prise en considération.

Les sorties

Elles font aussi une utilisation intensive des caractères de contrôle.

Les caractères de contrôle suivants sont ignorés par la routine de sortie de HAIFA qui les envoie tels quels à la sortie écran du moniteur: CTRL-D qui invoque le DOS, CTRL-G qui émet un bip et CTRL-à, CTRL-C, CTRL-I, CTRL-K, CTRL-L, CTRL-O, CTRL-Q, CTRL-U, CTRL-X, CTRL-crochet ouvrant (ESC), CTRL-

backslash, CTRL-crochet fermant, CTRL-%, CTRL-underscore qui sont ignorés par le moniteur.

Les caractères de contrôle suivants sont reprogrammés par les routines de sortie mais gardent la même signification que dans la routine COUT1 (la reprogrammation permet d'avoir le même effet dans le cas d'une sortie sur la page haute résolution): CTRL-H qui fournit un backspace, CTRL-J qui furnit un line fead et CTRL-M (RETURN) qui fournit un carriage return.

Les caractères suivants sont reprogrammés par la routine de sortie pour effectuer des actions particulières et sont ensuite envoyés à la routine de sortie du moniteur qui les ignore:
CTRL-A déplace le curseur à gauche comme CTRL-H, CTRL-S le déplace à droite du moniteur, CTRL-W le déplace vers le haut et CTRL-Z le déplace vers le bas comme CTRL-J.
CTRL-P efface l'écran comme HOME ou comme CALL - 936, CTRL-E efface jusqu'à la fin de la ligne comme CALL - 868 et CTRL-F efface jusqu'au bas de l'écran comme CALL - 958. Notons que dans le cas d'une sortie sur l'écran haute résolution ces caractères agissent de la même façon.

Les caractères suivants déterminent le mode de sortie des lettres: CTRL-N fait que toutes les lettres seront affichées telles qu'elles sont (mode par défaut), CTRL-R fait qu'elles seront toutes affichées comme minuscules, CTRL-B fait qu'elles le seront en minuscules et CTRL-Y fait que les majuscules seront affichées minuscules et vice-versa.

ATTENTION! Il convient de faire attention à la différence entre les minuscules à l'entrée et à la sortie car dans une entrée de ligne tout caractère entré est immédia- tement envoyé à la sortie. Donc si par exem- ple le mode CTRL-B de la sortie est mis, toutes les lettres entrées seront imprimées en minuscule, mais ce ne sont pas néces- sairement des minuscules du point de vue du buffer d'entrée.

& NORMAL PRINT fournit le mode normal d'entrée/sortie de caractères.

Ce mode est mis en place dès l'installation de HAIFA, ainsi que par les commandes d'initialisation graphique haute résolution ou par & INIT.

Les routines d'entrées/sorties correspondent alors exactement à ce qui est décrit ci-dessus avec écriture sur la page texte. Mais en ce qui concerne l'arrêt du listing (dans tout ce qui suit, on entend par listing le défilement des caractères à l'écran), CTRL-S ne marche plus, mais ESC est utilisé à la place, ce qui est plus commode. Et CTRL-C continue de stopper le listing.

& HGR PRINT fournit le mode graphique d'entrée/sortie de caractères.
L'option prime est la seule permise.

Tout ce qui a été décrit ci-dessus à propos des routines d'entrées/sorties reste encore valable, mais la sortie a lieu sur la page graphique haute résolution choisie avec écho sur la page texte pour permettre la relecture avec la flèche à droite. Il faut cependant se méfier de cette flèche à droite, car toute modification de l'écran texte (par un HOME, un CALL - 868, un CALL - 958, etc.) ou de la page graphique (par un & SHFT, un & CLR, etc.) fait que le caractère relu par cette flèche ne correspondra plus nécessairement à celui affiché sur la page graphique.

Notons donc que tous les caractères imprimés le sont de même sur la page texte, mais que les modes NORMAL, INVERSE et FLASH ne sont valables que pour cet écran texte, les caractères affichés à l'écran graphique utilisant le type décrit plus haut. D'ailleurs l'instruction & HGR PRINT remet en mode NORMAL et il est déconseillé de faire FLASH ou INVERSE tant que la sortie haute résolution a lieu.

Notons enfin que les commandes d'effacement (CTRL-E, F et P, et RETURN au clavier) remplissent les lignes qui sont effacées normalement sur la page texte avec la couleur haute résolution utilisée.

Parlons maintenant des caractères utilisés en haute résolution: on utilise une table de 768 ($300) octets, c'est-à-dire que l'on code chaque caractère dont le code ASCII est compris entre 32 et 127 par 8 octets (les caractères de contrôle ne sont pas codés car ils ne sont pas plus envoyés à l'écran graphique qu'à l'écran texte). Chaque octet représente le caractère par tranches horizontales de 7 bits, le 8ème bit jouant un rôle particulier. Voir l'instruction & TYPE = pour savoir comment ces bits sont envoyés sur la page graphique.

Montrons simplement à titre d'exemple comment peut être codée la lettre B:

octet 1	00011111	00011111
octet 2	00100010	00100010
octet 3	00100010	00100010
octet 4	00011110	00011110
octet 5	00100010	00100010
octet 6	00100010	00100010
octet 7	00011111	00011111
octet 8	00000000	00000000

Notons que la lettre est à l'envers car pour la page haute résolution, le bit de poids faible correspond au point le plus à gauche. Et l'on a mis ici le 8ème bit égal à 0.

L'interêt de cette sortie sur la page gra-

phique est de pouvoir générer ses propres caractères, latins, grecs, italiques, cyrilliques, gothiques, etc. Et l'on comprend qu'en remplaçant les lettres par des morceaux de dessins l'on puisse arriver à faire des dessins uniquement avec des ordres PRINT, ce qui peut se révéler plus agréable que les shapes. L'instruction & DRAW PRINT ci-dessous est cependant plus commode pour de tels dessins que & HGR PRINT qui fait des "carriage return" en bout de ligne et un scrolling en bas de page.

Remarquons que les fichiers de caractères utilisés par HAIFA sont totalement compatibles avec ceux de certains utilitaires du commerce, Applesoft ToolKit en particulier. Donc, pour avoir des caractères un peu particuliers, il suffit de donner à HAIFA les fichiers de caractères de Applesoft ToolKit.

& PDL LIST ([expression numérique]) ne modifie pas la routine d'entrée/sortie de caractères mais change le contrôle du listing.

L'expression, comprise entre 0 et 3, indique le numéro de la poignée de jeu (paddle) utilisée. La valeur de la poignée fournit la vitesse de défilement des caractères à l'écran, de 0 (très lent) à 255 (vitesse normale). Appuyer sur le bouton stoppe le listing, et lâcher le bouton le relance. N'oublions pas que le bouton 3 n'existe pas sur la prise d'entrée/sortie des manettes de jeu et donc si l'expression vaut 3, seule le potentiomètre de la poignée fonctionne.

CTRL-C continue de stopper le listing.

& STOP LIST ne change pas la routine d'entrées/sorties de caractères mais change le contrôle du listing.

Appuyer sur SHIFT-P stoppe l'envoi des caractères à l'écran, mais ils continuent de défiler de façon interne dans l'Apple II.

Appuyer sur une touche fait que le listing s'arrêtera quand il rencontrera le caractère correspondant à la touche (aussi bien en mode FLASH ou INVERSE que NORMAL). En fait, pour permettre cela, la routine arrête le listing sur tout caractère dont les six bits les plus faibles du code écran correspondent à ceux de la touche enfoncée. Ainsi taper sur RETURN arrêtera aussi bien le listing sur les M que sur les "carriage return", mais taper une lettre ne peut faire arrêter le listing sur la minuscule correspondante (mais le peut sur le caractère qui y correspond, ! pour a, " pour b ,etc.). Le manuel de référence Apple II fournit la liste des codes écran.

Appuyer alors sur une autre touche relance le listing de la même façon. Seule la touche ESC permet de le relancer définitivement sans arrêt. De plus cette touche permet à tout moment d'arrêter le listing.

Cette routine peut être très utile par exemple pour examiner des listings désassemblés ligne par ligne: il suffit alors d'appuyer sur RETURN à chaque ligne. De même, pour lire un texte mot à mot, il suffit d'appuyer sur la barre d'espacement à chaque mot. Enfin cette routine est très utile pour compter le nombre d'occurrences d'un caractère dans un programme.

Et CTRL-C stoppe encore le listing.

Notons que ces deux dernières instructions ne détruisent pas les routines d'entrée/sortie en vigueur, en prenant le contrôle des entrées/sorties avant le DOS à chaque caractère entré ou sorti, de façon à empêcher le DOS de reprendre en main les choses. Nous pouvons alors faire trois remarques: il est possible d'employer simultanément ces deux routines en plus de celles des entrées/sorties, mais il se pose peut-être un problème de stockage des registres internes du microprocesseur; il est aussi fortement conseillé de ne pas appeler deux fois de suite une de ces routines, car cela amène le système à "planter", la routine n'arrêtant pas de boucler sur elle-même; enfin, il faut signaler que les ordres DOS et certaines commandes Applesoft comme RUN déconnectent l'une ou l'autre de ces routines, mais pas les routines d'entrée/ sortie.

& DRAW PRINT fournit le mode graphique de dessins par entrée/sortie de caractères. L'option prime et l'option dièse sont permises.

Cette routine court-circuite la sortie de caractères et il faut donc absolument éviter de se retrouver au clavier dans ce cas (soit par enfoncement de CTRL-C, soit à cause d'une erreur dans le programme) car le système "plante" et le seul moyen de l'arrêter est d'appuyer sur RESET. On peut aussi prévenir plutôt que guérir en faisant & DRAW PRINT: PRINT CHR$(4)"PR#0", mais cela peut être dangereux (si le curseur se trouve en dehors de l'écran, on ne voit plus rien!) et il vaut mieux tout simplement mettre l'ONERR GOTO avec déconnection de & DRAW PRINT si une erreur vient à être rencontrée.

Ensuite en ce qui concerne la sortie de caractères, on n'utilise plus le curseur et la fenêtre de la page texte, mais un curseur spécial remis en haut à gauche par les initialisations haute résolution et par l'ordre & DRAW PRINT. Il n'y a donc pas d'écho des caractères sur la page texte. Ce curseur a la particularité de ne pas aller à la ligne et de ne pas faire de scrolling. Et quand il dépasse les limites de la fenêtre haute résolution (avec l'option dièse) ou de l'écran, on ne voit plus les caractères imprimés mais le curseur continue d'avancer horizontalement fictivement.

Les seuls caractères de contrôle tolérés sont ceux qui déplacent le curseur (CTRL-A,

CTRL-S, CTRL-W et CTRL-Z). Notons qu'ils agissent bien entendu même lorsque le curseur se trouve en dehors de la fenêtre. Les autres caractères de contrôle sont ignorés par la routine de sortie.

Notons au passage qu'à la différence de & HGR PRINT, cette instruction ne rétablit pas le mode normal et donc qu'il peut être conseillé de le faire si l'on se trouve en mode FLASH. Le mode INVERSE n'est pas gênant car il ne consiste qu'à mettre à 0 le 8ème bit du code ASCII, ce dont la routine de sortie de & DRAW PRINT se moque, mais le mode FLASH donne des résultats absurdes.

Il n'y pas de moyen d'arrêter le listing. Seul CTRL-C le stoppe, mais "plante" le système (voir ci-dessus).

De toute façon, cette instruction est destinée à faire des dessins en remplaçant les lettres dans les fichiers de caractères haute résolution par des morceaux de dessins, qu'on assemble en des gros dessins, en introduisant des CTRL-A, S, W et Z. Cela permet de faire des dessins de taille importante beaucoup plus facilement qu'en utilisant des shapes. Il suffit dans ce cas de faire un & POS (voir ci-dessous) et un ordre PRINT pour afficher le dessin.

& POS = ([expression 1]) , ([expression 2]) place le curseur haute résolution.

Les expressions sont numériques, la première étant comprise entre 1 et 40 (position horizontale) et la deuxième entre 1 et 24 (position verticale).

CONCLUSION

HAIFA est un programme certes intéressant, mais qui peut être amélioré et complété. On peut songer à introduire de nouvelles posibilités, comme le calcul avec des grands entiers (comme en PASCAL), des routines de tris, d'autres instructions graphiques, etc. Le tout étant d'avoir le temps: la version actuelle de HAIFA est le résultat d'un travail d'avril à juillet 82, et quelques mois de plus pourraient permettre de l'améliorer.

Une chose qui serait intéressante serait de pouvoir avoir les commandes de HAIFA à la carte de façon à ne pas utiliser trop de mémoire en ne retenant de HAIFA que ce qui sert dans un programme donné.

La rédaction de Pom's attend à ce sujet les suggestions et les propositions des lecteurs. Nous aimerions aussi que vous nous signaliez tous les bugs rencontrés en utilisant HAIFA car il en reste sûrement.

ANNEXE 1: LES MEMOIRES UTILISEES PAR HAIFA

L'occupation de la mémoire centrale

A. Version carte langage et Applesoft en ROM:

Partie 48 K
$0800 début du programme Applesoft
$8000 HIMEM et début de HAIFA
$9000 début de la table de shapes en haute résolution
$9500 début de la table de shapes en basse résolution
$9600 début du DOS

Carte langage: bank 1
$D000 début des tables de caractères HIRES (5 tables permises)
$DF00 buffer pour la routine & FRE

Carte langage: bank 2
$D000 début de la table des notes de musique

Carte langage: partie commune
$E000 début de HAIFA
$F000 fin de HAIFA
$F300 début de la pile (256 variables stockables)

Carte langage et ROM AUTOSTART:
$F800 moniteur

On voit que HAIFA utilise pleinement la capacité de l'Apple II muni de la carte langage: il y a à peine un peu plus de $300 octets libres avant $F300, d'autres octets pouvant être libérés par les diverses tables de HAIFA.

Mais nous avons aussi assemblé une version pour 48 K de HAIFA afin que tous puissent en profiter. Pour utiliser cette version, il suffit d'avoir 48 K et l'Applesoft en ROM ou l'INTEGER en ROM et l'Applesoft sur la carte langage.

B. Version 48 K:

$0800 début du programme Applesoft
$6800 HIMEM et début de HAIFA - 1ère partie
$7700 début de la table de shapes en basse résolution
$7800 début de HAIFA - 2ème partie
$8600 début de la table des notes de musique
$8C00 début de la table de shapes en haute résolution
$9000 début de la table de caractères HIRES (1 seule)
$9300 début du buffer de la routine & FRE
$9400 début de la pile (102 variables stockables)
$9600 début du DOS

Pour l'occupation du reste de la mémoire (pages graphiques, entrées/sorties, DOS,

etc.) voir les différents manuels de l'Apple II.

On en déduit donc que HAIFA est compatible avec certains utilitaires d'édition de programmes. Par exemple avec PLE (qui met la HIMEM en $9000) à condition de ne pas se servir de la partie de HAIFA comprise entre $9000 et $9500 - ou de transférer cette partie ailleurs (tables de shapes pour la version carte langage, caractères haute résolution, buffer de & FRE et pile pour la version 48 K).

HAIFA est compatible avec le APA de Applesoft ToolKit, à condition de charger celui-ci juste sous la HIMEM mise par HAIFA. En effet, APA est relogeable, car il utilise un "relocating loader". Notons toutefois que, dans ce cas, la HIMEM se retrouve environ en $7200 pour la version carte langage et $5A00 pour la version 48 K. La page graphique 2 est donc inutilisable dans ce dernier cas.

Les mémoires page zéro

HAIFA utilise pratiquement toutes les mémoires page zéro disponibles ($6 à 9, 18 et 19, 1E et 1F, EB à EF et FA à FF). Ces mémoires ne sont utilisées qu'à titre provisoire par HAIFA. Autrement dit n'importe quel utilisateur de HAIFA peut se servir des mémoires page zéro non utilisées par l'Applesoft ou le moniteur à condition qu'il ne fasse pas exécuter, quand il stocke des renseignements dans ces mémoires, des instructions HAIFA ou des entrées/sorties de caractères de HAIFA.

Les mémoires page 3

C'est dans la page 3 que HAIFA stocke les paramètres dont il a besoin. Les mémoires suivantes ne pourront donc pas être utilisées dans un programme qui se sert de HAIFA. Si l'on a absolument besoin de la page 3, on peut assembler HAIFA en mettant ses mémoires de stockage ailleurs.

$300 (768) STACK
$301 (769) ----- pointeur du sommet de la pile (pointe le début de la pile après l'initialisation)
$302 (770) STKLNG longueur de la pile
$303 (771) STKPTR pointeur de la pile
$304 (772) LRSHAPE
$305 (773) ------- adresse du début de la table de shapes LORES
$306 (774) ASCII
$307 (775) ----- adresse du début de la table de caractères HIRES (adresse de la première de ces tables après l'initialisation)
$308 (776) MUSIC
$309 (777) ----- adresse de début de la table des notes de musique
$30F (783) PADDLE numéro de la poignée de jeu utilisée par & PDL PRINT
$310 (784) MODAND premier argument du MODE
$311 (785) MODOR second argument du MODE
$312 (786) MODEOR 3ème argument du MODE
$313 (787) LFT limite gauche de la fenêtre
$314 (788) RGT limite droite de la fenêtre
$315 (789) TOP limite supérieure de la fenêtre
$316 (790) BTM limite inférieure de la fenêtre
$317 (791) TRTLROT angle de la tortue
$318 (792) TYPE type
$319 (793) HRCH position horizontale pour & DRAW PRINT
$31A (794) HRCV position verticale pour & DRAW PRINT
$31B (795) PRPAG vaut 0 ou 128
(128 si l'option prime est mise pour & HGR PRINT ou & DRAW PRINT, 0 si elle ne l'est pas)
$31C (796) PRWND vaut 0 ou 128
(128 si l'option dièse est mise pour & DRAW PRINT, 0 si elle ne l'est pas)
$31D (397) PROUT vaut 0 ou 128
(0 si la sortie de caractères est celle de & NORMAL PRINT et 128 pour & HGR PRINT)
$31E (398) MINFLAG drapeau majuscules/minuscules pour la sortie
(est utilisé par & HGR PRINT et & NORMAL PRINT: 0 normal, 1 inverse minuscules et majuscules, 2 force en minuscules et 3 force en majuscules)
$31F (399) CAPIT drapeau majuscules/minuscules pour l'entrée
(pour & HGR PRINT et & NORMAL PRINT; 0 majuscules, 32 minuscules et 128 si ESC vient d'être enfoncé)
$320 (400) CHRSTRE stocke le caractère à l'entrée
(pour & HGR PRINT et & NORMAL PRINT; est utilisé pour tromper la routine GETLN qui force le mode minuscule quand on rentre une ligne)
$321 (401) RGTARW drapeau majuscules/minuscules pour la flèche à droite (pour la relecture en entrée, cf. MINFLAG)
$322 (402) MUSFLAG drapeau pour CTRL-Q dans la routine d'entrée.

Bien entendu, l'utilisateur de HAIFA est libre de POKEr lui-même ces mémoires comme il l'entend. C'est surtout utile quand on veut changer les adresses des tables utilisées.

ANNEXE 2 : LES FICHIERS FOURNIS SUR LES DISQUETTES

HAIFA.TXT & RAMCARD.TXT sont les deux fichiers sources qui constituent HAIFA. Ils sont écrits pour l'assembleur LISA 2.5 et sont donc malheureusement incompatibles avec les autres assembleurs, car on ne peut les BLOADer. Il y a assez peu de modifications à faire pour avoir les fichiers 48 K (HAIFA.TXT 1 et HAIFA.TXT 2): changer les ORG, ICL, DCM et les valeurs initiales des mémoires de la page 3; il faut aussi supprimer les BIT avec

les adresses de lecture et écriture de la carte langage (cela n'est pas nécessaire si l'on a l'Applesoft en ROM, mais c'est indispensable si l'on a l'INTEGER en ROM et l'Applesoft sur la carte langage).

Faute de place, les fichiers HAIFA.TXT et RAMCARD.TXT ne sont pas fournis sur la disquette d'accompagnement de Pom's, qui est déjà pleine, mais sur une disquette spéciale contenant ceux-ci dans les versions 64K et 48K (plus de 150 secteurs par version ...). A propos de ces fichiers sources, notons qu'ils ne sont pas faciles à éplucher: ils sont en effet si gros qu'il a fallu retirer les commentaires (les seuls qui restent sont les titres des routines), et d'autre part, comme ils ont saturé la table des étiquettes de l'assembleur, les adresses des routines Applesoft et moniteur sont données la plupart du temps en hexadécimal plutôt qu'avec une étiquette.

Notons aussi que la version 48 K n'est pas du tout optimisée au point de vue de la place en mémoire car ce n'est que la transposition de la version 64 K, pour laquelle l'assemblage de routines sur la carte langage n'a pour but, aux dépens de la place totale utilisée, que de réduire la place prise dans la partie 48 K. La disquette HAIFA SOURCE est vendue en supplément au prix habituel de 50 francs TTC.

HAIFA.CODE et RAMCARD.CODE sont les fichiers obtenus dans la version pour carte langage. Attention au fait que le code objet de chacun d'eux a été généré à partir de l'adresse $800: il faut donc faire un BLOAD ,A$8000 pour le premier et , A$E000 pour le second.

HAIFA.CODE 1 et HAIFA.CODE 2 sont les fichiers codes de la version 48 K. On peut faire la même remarque que précédemment: ,A$6800 pour le premier et ,A$7800 pour le second.

MAKE HAIFA.EXEC est un utilitaire qui permet de créer un fichier EXEC nommé HAIFA.EXEC qui contient des lignes de programmes destinées au chargement de HAIFA et de ses fichiers: ce programme demande quels sont les fichiers à charger parmi les tables de shapes, caractères et de musique, avec les adresses de début si ce ne sont pas celles par défaut; il demande alors le numéro de la ligne de départ pour les instructions de chargement et l'incrément pour passer d'une ligne à la suivante. Il ne lui reste plus alors qu'à créer le fichier HAIFA.EXEC.

Une fois celui-ci créé, il suffit de faire EXEC HAIFA.EXEC pour introduire dans le programme Applesoft les instructions de chargement. Attention simplement à ne pas avoir de numéro de ligne commun entre celles du programme et celles apportées par le fichier HAIFA.EXEC, sous peine de faire disparaître celles du programme. On peut faire l'instruction EXEC avant de taper le programme Applesoft récepteur ou pendant qu'on le tape ou encore après l'avoir tapé.

Notons, pour ceux qui veulent faire "à la main" les instructions de chargement, qu'il faut charger les tables avant les fichiers codes car ceux-ci ont été automatiquement sauvés à l'assemblage avec une longueur de $1000, plus grande que la longueur réellement utilisée. D'autre part, si jamais on termine le chargement par celui de la première partie du code (HAIFA.TXT ou HAIFA.TXT 2), il ne faut pas faire un BRUN, mais un BLOAD et un CALL: c'est un bug bien connu du DOS: il ne faut pas BRUNer les programmes qui changent les entrées/sorties.

HAIFA.HGR SHAPES est une table de formes haute résolution contenant les caractères ASCII de 33 à 95, qu'il est conseillé d'utiliser avec XDRAW et avec SCALE=1. Le numéro de la forme est le caractère ASCII moins 32.

HAIFA.CHARSET est une table de caractères haute résolution contenant les caractères ASCII standards. Pour son format, voir l'instruction & HGR PRINT.

Les autres programmes fournis sur la disquette sont des programmes de circonstance destinés au générique de la disquette de Pom's numéro 5. Il sont repérés par le préfixe Pom's.

ANNEXE 3 : LES REFERENCES

HAIFA s'est inspiré de Call-Apple In Depth (All About Applesoft) pour les routines suivantes: & AT et & TO, & GOTO et & GOSUB, & SWAP, & CLEAR, & DEL, & INPUT, & CALL, & NEW, & PRINT USR et & REPT PRINT.

HAIFA a utilisé NIBBLE EXPRESS 1 et 2 pour les instructions suivantes: & FRE, & STORE et & RECALL, & DUMP.

HAIFA a imité la ROM Applesoft pour réaliser les routines suivantes: & PEEK et & POKE, & IF et & ELSE, & X HPLOT, & N HPLOT et & HPLOT, & ONERR, & DRAW et & XDRAW, & GET et & INPUT.

L'instruction & POP est tirée d'un jeu de la revue SOFTSIDE et & NOT est tout-à-fait classique.

L'instruction & PLAY1 est tirée de programmes diffusés par des groupements américains et & PLAY2 est la routine de Electric Duet.

L'idée de menu (& LET) a été ispirée par l'article envoyé à Pom's par Denis SUREAU et publié dans ce numéro.

Les autres instructions, c'est-à-dire essentiellement les puissantes instructions graphiques, ont été conçues et réalisées spécialement pour HAIFA.

"Croque moi

et tu inventeras!!

L'homme invente de plus en plus. Et de mieux en mieux.
Pour repousser les limites du possible, il s'est fabriqué des outils à sa mesure.
L'ordinateur personnel Apple en est un.

Rappelez-vous. Il n'y a pas si longtemps, l'ordinateur personnel c'était un rêve. Aussi fou que de vouloir posséder son propre vaisseau spatial.

Et puis il y eut Apple.

L'informatique indépendante, abordable (un Apple coûte moins qu'un simple photocopieur) et accessible (on apprend à s'en servir en quelques heures).

Un Apple ne vient jamais seul. Avec lui, vous disposez d'une bibliothèque de programmes avec lesquels vous pouvez vous mettre tout de suite au travail. Sans avoir à apprendre le langage informatique.

Un Apple, c'est le meilleur moyen d'aller plus vite et plus loin. D'être créatif sans aucune contrainte. De regagner le temps perdu en tâches répétitives, en routine. D'aller jusqu'au bout de chaque nouvelle idée. De redevenir inventif à 100 %.

Vous faut-il d'autres bonnes raisons?

Alors, examinez une de vos journées de travail, vous en trouverez. Mais si vous savez déjà qu'un ordinateur personnel peut vous faire du bien, documentez-vous (voyez le bon à croquer Apple au bas de cette page).

Et gardez bien en tête que votre ordinateur personnel doit disposer de programmes pour vos travaux habituels. Et qu'il doit être capable de grandir en fonction de vos besoins (Apple dispose de plus d'accessoires que n'importe quel autre ordinateur personnel).

Choisissez aussi un ordinateur célèbre et qui a fait ses preuves : 400.000 Apple fonctionnent chaque jour dans le monde. C'est la meilleure preuve de leurs hautes performances et la certitude d'un service disponible sur le champ. On ne devient pas célèbre par hasard.

Un Apple, c'est vrai, change les façons de travailler, de penser, de décider.

A vous de décider.

BON A CROQUER

SEEDRIN
Avenue de l'Océanie Z.I. de Courtabeuf
91944 LES ULIS CEDEX

☐ Que lire? Si vous voulez vous familiariser avec le monde de l'ordinateur personnel à travers la littérature Apple et les revues spécialisées, cochez cette case.

☐ En français ☐ En anglais

☐ Si vous ne pouvez plus attendre, cochez cette case pour recevoir la liste des revendeurs agréés Apple.

Nom _____

Société _____

Adresse _____

_____ Code postal _____

apple
l'ordinateur personnel

Création graphique en Pascal

Michel Crimont

Le Pascal offre de grandes possibilités de graphisme haute résolution; malheureusement, l'écriture des blocs utilisés par la procédure DRAWBLOCK est encombrante dans un programme et les dessins ne peuvent être utilisés que par le programme qui les crée, au fur et à mesure des besoins.

D'où l'idée de créer un fichier contenant des blocs prédessinés du type SYSTEM.CHARSET mais plus grands, que tout programme pourra venir lire pour en utiliser les graphismes.

Le dessin se fera sur une grille de grande taille, chaque point H.R. étant représenté par un point de 8x8 que l'on peut allumer ou éteindre sur l'écran et qui, déplacé par les touches I-J-K-M, dessine les graphismes. Le dessin terminé est réaffiché dans sa taille réelle avant l'enregistrement. Il sera par ailleurs possible de relire 36 figures simultanément à l'écran pour choisir celles convenant à un programme donné.

Les déclarations

-Const TAILLE=14: les figures font 15x15, ce qui les rend faciles à tracer et n'empêche pas de les réunir par groupes de 10 pour de plus grands dessins.

-Type:
FORME=PACKED.ARRAY[0..taille,0..taille] of booléen: image haute résolution
CHOIDECA est un set de caractère
POINT=PACKED.ARRAY[0..7,0..7] of Booléen est simplement le point grossi nécessaire pour dessiner.

Les variables

-IMAGE : variable de type FORME
-ALBUM : fichier d'images. Les caractères sont ceux de la librairie courante pour les caractères de contrôle d'écran
-NOM est celui du fichier contenant les images
-DOT1, DOT2 et DOTN sont les points nécessaires pour dessiner
-INDEX est le nombre d'images du fichier.

Les procédures

Les procédures MESSAGE, PRENRETURN, PRENCAR, PRENCHAINE, OUI et SELECT sont des procédures courantes, déjà décrites dans les précédents numéros de Pom's.

INIT initialise les caractères de contrôle d'écran et les variables suivantes qui serviront à dessiner:
DOT1 : point de positionnement;
DOT2 : point blanc;
DOTN : point noir.

NOMFICHIER prend le nom du fichier image au clavier. Si le numéro du drive, 4 ou 5, n'est pas précisé, ajoute 4 et place systématiquement le suffixe .PIC après le nom du fichier. La procédure essaie ensuite d'ouvrir le fichier. Si elle le trouve, elle le lit en incrémentant l'index qui se trouve donc sur la dernière image libre si l'on veut continuer à écrire dans un fichier préexistant. Si elle ne le trouve pas, le fichier est créé.

GRILLE trace sur l'écran une grille de NX*NY de côté avec un espacement entre les "barreaux" de PASX et PASY.

MARQUE place le point DT (noir ou blanc) aux coordonnées X,Y.

ECRIMAGE est la procédure de dessin proprement dite. Appel de NOMFICHIER : on prend le nom du fichier et on l'ouvre ou on le crée, ce qui en même temps positionne INDEX sur un endroit libre du fichier. La boucle REPEAT permet ensuite de créer successivement plusieurs dessins dans cette boucle. A chaque fois:
.on efface l'écran (INITTURTLE)
.on trace la grille (GRILLE)
.on donne les indications pour dessiner: déplacement du curseur avec I-J-K-M pour haut-gauche-droite-bas, T pour Trace, O pour pas de Trace (effacer) et F (fin).

On place le point en 0,0 c'est-à-dire en bas et à gauche de la grille, et l'on entre dans une autre boucle REPEAT jusqu'à ce que l'on tape F pour fin. Dans cette boucle, on trace le point en cours, soit DOT2 et IMAGE (Y.X)=Vrai si TRACE est vrai, soit DOTN et IMAGE (Y.X)=Faux si TRACE est faux - ce qui assure la correspondance entre l'écran et le tableau de booléens représentant l'image.

Le CASE DIR permet de reconnaître les diverses commandes :
I pour monter (maximum TAILLE)
K pour déplacer à droite (maximum TAILLE)
M pour descendre (minimum 0)
J pour déplacer à gauche (minimum 0)

T place Trace à vrai
O place Trace à faux

La procédure DRAWBLOCK, une fois le "F" tapé, affiche l'image en vraie grandeur pour la contrôler et, si on le désire, l'enregistrer; il est ensuite possible d'interrompre l'écriture ou de continuer le processus avec un autre dessin.

AFFICHE est utilisé en lecture. La variable IND numéro de l'image dans le fichier sert à positionner l'image sur l'écran dans l'une des 36 cases par calcul de POSY et POSX. IND est ensuite transformé en chaîne alpha et écrit comme numéro de l'image par le WSTRING. L'image elle-même est placée au numéro calculé par la procédure DRAWBLOCK.

LIRIMAGE appelle de façon itérative AFFICHE pour afficher chaque fois une page de 36 images. Pour cela, comme pour ECRIMAGE, cette routine demande le nom du fichier que l'on désire. INDEX est remis à 0 et le programme affiche successivement la grille et les 36 images munies de leur numéro.

Le corps du programme sélectionne simplement la fonction désirée : lecture, écriture ou arrêt.

Utilisation dans un programme Pascal

Le programme Pascal devra comporter en déclaration
USESTURTLEGRAPHICS
CONST TAILLE=14
TYPE FORME
et VAR ALBUM FILE OF FORME

Le fichier ALBUM ayant été créé préalablement sous le nom souhaité, par exemple APLSET, le fichier APLSET.PIC doit être placé sur le même disque que le programme qui doit l'utiliser.

Ce programme pourra donc simplement ouvrir le fichier et lire le caractère voulu, par exemple le sixième par:

```
RESET(ALBUM,"*APLSET.PIC")
SEEK(ALBUM,6),GET(ALBUM)
DRAWBLOCK(ALBUM ,....etc.)
```

ou éventuellement définir préalablement un tableau, par exemple :

```
EXPO:ARRAY[1..26] OF FORME
```

tableau qui sera rempli par lecture de l'album

```
SEEK(ALBUM,1)
FOR I:=1 TO 26 DO
Begin
GET(ALBUM);EXPO[I]:=ALBUM
End
```

Il suffira ensuite d'utiliser EXPO[X] dans la procédure DRAWBLOCK. Nous attendons avec impatience vos créations les plus réussies !

logma

Une informatique de gestion adaptée aux besoins des gestionnaires et réalisée par des gestionnaires.

ÉTUDIE
- opportunité d'utilisation de l'outil micro-informatique
- intégration entre informatique traditionnelle et personnelle
- politique de la communication dans l'entreprise

FORME
- formation à l'utilisation de la micro-informatique

RÉALISE
- réalisation de programmes à la demande

LIVRE
- livraison de systèmes clés en main, avec des progiciels de GESTION DE STOCK, PAYE, COMPTABILITE.

Nous sommes gestionnaires avant d'être informaticiens. L'informatique doit s'adapter à l'homme, et non l'inverse.
L'outil micro-informatique répond particulièrement bien à ce souci de qualité et d'efficacité du travail, dans des conditions conviviales.
Nombreuses références en informatique traditionnelle - divers matériels - et en informatique individuelle - principalement Apple - auprès des PME et des groupes industriels.

logma s.a. Centre La Châtaigneraie - 29, avenue de Versailles - 78170 La-Celle-St-Cloud - Tél : (3) 918.13.07

```
(* PROGRAMME DESTINE A DESSINER DES BLOCS 15*15 EN H.R. *)
(*$C (C) M.CRIMONT 20/08/80 *)
(*$S+*)
PROGRAM BASEGRAPHE;

USES TURTLEGRAPHICS;

CONST   TAILLE      =14;

TYPE    FORME       =PACKED ARRAY[0..TAILLE,0..TAILLE] OF BOOLEAN;
        CHOIDECA    =SET OF CHAR;
        POINT       =PACKED ARRAY[0..7,0..7] OF BOOLEAN;

VAR     IMAGE       :FORME;
        COM,HOME,
        BS,EOL,SON,
        INV,NORM,CR:CHAR;
        NOM         :STRING;
        ALBUM       :FILE OF FORME;
        DOT1,DOT2,
        DOTN        :POINT;
        INDEX       :INTEGER;


(*****************************************)
(* PROCEDURES HABITUELLES COURANTES     *)
(*****************************************)

PROCEDURE MESSAGE(X:INTEGER;S:STRING);
BEGIN
  GOTOXY(0,X);WRITE(S,EOL)
END;

PROCEDURE PRENRETURN;
VAR SORT: CHAR;
BEGIN
  REPEAT
    READ(KEYBOARD,SORT);
  UNTIL EOLN(KEYBOARD)
END;

FUNCTION  PRENCAR(BONSET:CHOIDECA):CHAR;
VAR CH:CHAR;
   BON:BOOLEAN;
BEGIN
  REPEAT
    READ(KEYBOARD,CH);
    IF EOLN(KEYBOARD) THEN CH:=CR;
    BON:=CH IN BONSET;
    IF NOT BON THEN WRITE(CHR(7))
      ELSE IF CH IN[' '..'^'] THEN WRITE(CH);
  UNTIL BON;
PRENCAR:=CH
END;

PROCEDURE PRENCHAINE(LONGMAX:INTEGER;BONSET:CHOIDECA;VAR S:STRING);
VAR S1:STRING[1];
    CONT:STRING;
BEGIN
  S1:=' ';
```

```
  CONT:='';
  REPEAT
    IF LENGTH(CONT)=0 THEN S1[1]:=PRENCAR(BONSET)
      ELSE IF LENGTH(CONT)=LONGMAX THEN S1[1]:=PRENCAR([CR,CHR(8)])
             ELSE S1[1]:=PRENCAR(BONSET+[CR,CHR(8)]);
    IF S1[1] IN BONSET THEN CONT:=CONCAT(CONT,S1)
      ELSE IF S1[1]=CHR(8) THEN
        BEGIN
          WRITE(BS,' ',BS);
          DELETE(CONT,LENGTH(CONT),1)
        END;
  UNTIL S1[1]=CR;
  S:=CONT
END;

FUNCTION OUI:BOOLEAN;
BEGIN
  OUI:=PRENCAR(['O','N']) IN ['O']
END;

FUNCTION SELECT(X:INTEGER;S:STRING;BON:CHOIDECA):CHAR;
BEGIN
  MESSAGE(X,S);SELECT:=PRENCAR(BON)
END;

(*****************************************)
(* INIT DU PROGRAMME ET CREATION D'UN    *)
(* POINT BLANC ET D'UN NOIR              *)
(*****************************************)
PROCEDURE INIT;
VAR I,J   :INTEGER;
BEGIN
  HOME:=CHR(27);  EOL:=CHR(29);  BS  :=CHR(8);
  SON :=CHR(7);   INV:=CHR(18);  NORM:=CHR(20);  CR:=CHR(13);
  FOR I:=0 TO 7 DO
    FOR J:=0 TO 7 DO
    BEGIN
      DOT1[I,J]:=FALSE;DOT2[I,J]:=TRUE;
    END;
    DOTN:=DOT1;
  FOR I:=2 TO 5 DO FOR J:=2 TO 5 DO DOT1[I,J]:=TRUE;
  FOR I:=3 TO 4 DO FOR J:=3 TO 4 DO DOT1[I,J]:=FALSE;
END;

(*****************************************)
(* OUVRE LE FICHIER ALBUM                *)
(*****************************************)
PROCEDURE NOMFICHIER;
VAR BON:BOOLEAN;
BEGIN
  REPEAT
    PAGE(OUTPUT);
    CLOSE(ALBUM);
    BON:=TRUE;INDEX:=0;
    REPEAT
      MESSAGE(10,'NOM DU FICHIER:');
      PRENCHAINE(13,['0'..'9','A'..'Z','#',':'],NOM);
    UNTIL NOM<>'';
    IF POS('#',NOM)=0 THEN NOM:=CONCAT('#4:',NOM);
    IF LENGTH(NOM)>13 THEN NOM:=COPY(NOM,1,13);
```

```
      NOM:=CONCAT(NOM,'.PIC');
      (*$I-*)
      RESET(ALBUM,NOM);
      BON:=IORESULT=0;
      IF BON THEN
      BEGIN
        SEEK(ALBUM,INDEX);GET(ALBUM);
        WHILE NOT EOF(ALBUM) DO
        BEGIN
          INDEX:=INDEX+1;
          GET(ALBUM)
        END
      END ELSE
      BEGIN
        REWRITE(ALBUM,NOM);
        BON:=IORESULT=0
      END;
      IF NOT BON THEN
        BEGIN
          WRITE(HOME);
          GOTOXY(0,10);WRITE(SON,'TITRE OU LECTEUR NON VALIDE');
          WRITE('     FAITES, <RETURN> ');PRENRETURN;
          CLOSE(ALBUM)
        END
      (*$I+*)
    UNTIL BON
  END;

  (*****************************************)
  (* TRACE UNE GRILLE DE NX*NY AVEC LES    *)
  (* PAS PASX,PASY                         *)
  (*****************************************)
  PROCEDURE GRILLE(NX,NY,PASX,PASY:INTEGER);
  VAR I     :INTEGER;
  BEGIN
    INITTURTLE;
    FOR I:=0 TO NX+1 DO
      BEGIN
        PENCOLOR(NONE);
        MOVETO(I*PASX,NY*PASY+PASY);
        PENCOLOR(WHITE);
        MOVETO(I*PASX,0)
      END;
    FOR I:=0 TO NY+1 DO
      BEGIN
        PENCOLOR(NONE);
        MOVETO(0,I*PASY);
        PENCOLOR(WHITE);
        MOVETO(NX*PASX+PASX,I*PASY)
      END
  END;

  (*****************************************)
  (* PLACE UN POINT BLANC OU NOIR EN X,Y *)
  (*****************************************)
  PROCEDURE MARQUE(DT:POINT;X,Y:INTEGER);
  VAR TON   :INTEGER;
  BEGIN
    DRAWBLOCK(DT,2,0,0,7,7,X*10+1,Y*10+1,10)
  END;
```

```
(*****************************************)
(* PROCEDURE PRINCIPALE D'ECRITURE       *)
(* DES DESSINS SOUHAITES                 *)
(*****************************************)
PROCEDURE ECRIMAGE;
VAR X,Y        :INTEGER;
    DIR        :CHAR;
    TRACE      :BOOLEAN;
BEGIN
  NOMFICHIER;
  REPEAT
    FOR X:=0 TO TAILLE DO FOR Y:=0 TO TAILLE DO IMAGE[X,Y]:=FALSE;
    INITTURTLE;
    GRILLE(TAILLE,TAILLE,10,10);
    PENCOLOR(NONE);MOVETO(0,181);PENCOLOR(WHITE);
    WSTRING('DEPLACER LE CURSEUR AVEC LES TOUCHES');
    PENCOLOR(NONE);MOVETO(0,171);PENCOLOR(WHITE);
    WSTRING('I=HAUT K=DROITE M=BAS J=GAUCHE');
    PENCOLOR(NONE);MOVETO(0,161);PENCOLOR(WHITE);
    WSTRING('F=FIN  T=TRACE  O=NOIR');
    X:=0;Y:=0;MARQUE(DOT1,X,Y);TRACE:=FALSE;
    REPEAT
      DIR:=PRENCAR(['I','K','M','J','F','T','O']);
      IF TRACE THEN
        BEGIN
          MARQUE(DOT2,X,Y);
          IMAGE[Y,X]:=TRUE
        END ELSE
        BEGIN
          MARQUE(DOTN,X,Y);
          IMAGE[Y,X]:=FALSE
        END;
      CASE DIR OF
        'I':BEGIN
              Y:=Y+1;IF Y>TAILLE THEN Y:=TAILLE;MARQUE(DOT1,X,Y)
            END;
        'K':BEGIN
              X:=X+1;IF X>TAILLE THEN X:=TAILLE;MARQUE(DOT1,X,Y)
            END;
        'M':BEGIN
              Y:=Y-1;IF Y<0 THEN Y:=0;MARQUE(DOT1,X,Y)
            END;
        'J':BEGIN
              X:=X-1;IF X<0 THEN X:=0;MARQUE(DOT1,X,Y)
            END;
        'T':TRACE:=TRUE;
        'O':TRACE:=FALSE
      END
    UNTIL DIR='F';
    DRAWBLOC(IMAGE,2,0,0,15,15,230,75,10);
    PENCOLOR(NONE);MOVETO(160,50);PENCOLOR(WHITE);
    WSTRING('ENREGISTRER (O/N) ');
    IF OUI THEN
      BEGIN
        SEEK(ALBUM,INDEX);ALBUM^:=IMAGE;PUT(ALBUM);INDEX:=INDEX+1
      END;
    MOVETO(160,50);WSTRING('CONTINUER ? (O/N)
  UNTIL NOT OUI;
  CLOSE(ALBUM,LOCK)
END;
```

```
(*****************************************)
(* PLACE UN DESSIN DANS LA GRILLE        *)
(*****************************************)
PROCEDURE AFFICHE(DESSIN:FORME;IND:INTEGER);
VAR POSX,POSY,I:INTEGER;
    NUM         :STRING;
BEGIN
  I:=IND MOD 36;
  POSY:=150-((I DIV 6)*30);POSX:=(I MOD 6)*45;
  STR(IND,NUM);
  PENCOLOR(NONE);MOVETO(POSX+5,POSY+20);PENCOLOR(WHITE);
  WSTRING(NUM);
  DRAWBLOC(DESSIN,2,0,0,15,15,POSX+15,POSY+5,10)
END;

(*****************************************)
(* LECTURE D'UN FICHIER EXISTANT         *)
(*****************************************)
PROCEDURE LIRIMAGE;
BEGIN
  NOMFICHIER;INDEX:=0;
  MESSAGE(22,'TAPER <RETURN> POUR LA GRILLE SUIVANTE');PRENRETURN;
  (*$I-*)
  GRILLE(6,6,45,30);
  SEEK(ALBUM,INDEX);GET(ALBUM);
  WHILE NOT EOF(ALBUM) DO
  BEGIN
    AFFICHE(ALBUM^,INDEX);INDEX:=INDEX+1;
    IF (INDEX MOD 36)=0 THEN
      BEGIN
        PRENRETURN;
        GRILLE(6,6,45,30)
      END;
    SEEK(ALBUM,INDEX);GET(ALBUM)
  END;
  PRENRETURN
END;

(*****************************************)
(* PROGRAMME PRINCIPAL                   *)
(*****************************************)

BEGIN
  INIT;
  REPEAT
    PAGE(OUTPUT);
    TEXTMODE;
    COM:=SELECT(0,'BASEGRAPHE: L(IRE, E(CRIRE, Q(UITTER ? ',['L','E',
    CASE COM OF                                                    'Q']);
      'L':LIRIMAGE;
      'E':ECRIMAGE
    END;
  UNTIL COM='Q'
END.
```

CX Multigestion à l'essai

Hervé Thiriez

Objectifs du programme

Le programme CX Multigestion est à de nombreux titres un produit français fort original. Nous sommes tentés de reprendre, en pensant au système M/DOS (voir numéro 4) ou à CX Multigestion, avec un ton parodiaque non dissimulé "En France, on n'a pas de microprocesseur, mais on a des idées ...".

Le logiciel CX Multigestion est une base de données qui cache son nom; c'est d'ailleurs un produit hybride qui possède de nombreuses caractéristiques d'une base de données, mais pas toutes, et qui a été conçu dans une optique légèrement différente. Ce produit est le premier d'une famille dont le but est de former un ensemble cohérent suffisant pour aider à gérer, sans faire appel à d'autres logiciels, une PME/PMI. Cette ambition va jusqu'à l'intégration de traitement de texte en relation avec les fichiers définis et utilisés par la programme CX-Multigestion.

En quelques mots, CX Multigestion vous permet de définir des fiches, d'opérer des recherches, des tris et des classements, d'effectuer automatiquement certains calculs sur les fiches, ou à travers les fiches sélectionnées, de fusionner des fichiers construits sur une base commune, et enfin de sortir des tableaux (écran ou imprimante) ou des étiquettes.

Création des fiches

En quelques mots, ce logiciel permet de définir des fiches ayant un certain nombre de zones, chacune étant définie comme alphanumérique, numérique ou date. Les différences principales par rapport à la définition de zones dans la plupart des bases de données sont les suivantes :

-la définition d'une zone "date" est assez flexible : 12 FEVRIER 1982, 12 FEV 82, 12 F 82, 12 2 82, 12/2/82, 12-02-82, 12 2 ou 12 F (année définie par défaut) étant des représentations reconnues et acceptées.

-la définition d'un nombre possède aussi une flexibilité originale avec la reconnaissance des formes suivantes : 1002.25 ou 1002,25 ou 1002.25 ou 1002,25 ou 1002,25 F. Par contre, et là aussi la limitation revêt une certaine originalité, les décimales au-delà de la seconde sont ignorées.

-il n'y a pas à définir de longueur maximale de zone : celle-ci est limitée d'office à 24 caractères. Il est par contre possible d'affecter une limite inférieure à la longueur d'une zone.

-les libellés sont obligatoirement limités à une longueur de 11 caractères, car une ligne d'écran est réservée pour chaque zone (enregistrement + texte + présentation).

Sur le plan technique, CX Multigestion gère lui-même tous ses pointeurs, et n'a pas besoin de réserver une longueur fixe par zone : les longueurs sont variables et la place occupée est directement fonction de l'information placée dans la zone.

On distingue deux types de dossiers :

1- les "dossiers libres" formés de fiches libres correspondant à celles que l'on définit dans les bases de données habituelles, sous réserve des remarques que nous venons de faire dans les paragraphes précédents. On peut définir 45 zones par fiche, au rythme de 15 par écran; le quatrième et dernier écran (feuille de calcul) est réservé à la définition de formules internes à la fiche et reliant diverses zones entre elles.

2- les "dossiers nominatifs" comportent aussi trois écrans de données et un écran jouant le rôle de feuille de calcul. Le premier écran possède une structure prédéfinie, avec des rubriques classiques : M/STE, NOM, PRENOM, ADR1, ADR2, CP-V, TEL1, TEL2 et trois lignes de notes.

Pour tous les dossiers, libres ou nominatifs, un numéro de fiche est automatiquement attribué par le programme, par incrémentation de 1 à chaque création.

Recherche d'un dossier

La recherche d'un dossier précis peut s'effectuer de deux façons différentes, dont le choix est laissé à l'utilisateur : soit à partir du numéro de la fiche (fourni par le programme), soit à partir de la première et de la deuxième clef de recherche (nom et prénom, pour un dossier nominatif). Une modification des paramètres du programme autorise, dans le cas d'un fichier libre, de considérer la première clef (ou la seconde) seule pour la recherche.

Deux inconvénients à ce mode de recherche :

1- lors de la création, le programme ne vérifie pas s'il existe déjà une fiche ayant les mêmes deux clefs (ou les mêmes nom et prénom, pour un fichier libre) que la fiche en cours de création. Il y a par conséquent risque de problème en création, mise à jour ou recherche.

2- le nom et le prénom doivent être écrits en entier, ce qui augmente sensiblement le risque d'erreur ou de perte de temps dans le cas des noms longs ou à représentations potentielles multiples (plus les problèmes dus aux fautes d'ortographe) : ainsi JEAN DUPONT de sera pas reconnu dans la recherche de JEAN DUPOND, alors que le problème pourrait être résolu par l'utilisation d'un caractère "wild card" avec par exemple JEAN DUPON?, le "?" pouvant remplacer tout caractère. De même, une recherche sur J-F DUVIVIER (pour citer une célébrité locale) ne donnera rien si l'on a créé une fiche J.-F. DUVIVIER.

Une autre solution consiste à passer par un dossier de traitement pour effectuer la recherche.

Sélection de fiches

Nous sommes franchement plus positifs à l'égard de la façon dont CX Multigestion gère la sélection de fiches. Il est possible de sélectionner des fiches à partir de plusieurs clés simultanément, la sélection se réalisant à travers les modes suivants :

- égalité, à partir de la gauche pour les zones alphanumériques : "BA" sélectionnera toutes les fiches dont le texte dans la zone considérée débute par les lettres "BA"
- par analogie ou par mot-clef à l'intérieur d'une zone
- par différence (inverse du mode de sélection par égalité)
- avec indication de limites, qu'il s'agisse de limites de dates, numériques ou alphanumériques.

Ces sélections sont superposables, le nombre total de critères de sélection devant rester inférieur ou égal à 12. Ainsi, on peut sélectionner toutes les fiches pour lesquelles le nom débute par une lettre de P à S, mais cela comptera pour deux critères sur les douze autorisées. Les numéros des fiches gardées au terme d'une sélection peuvent être mémorisés et servir de point de départ à une seconde et dernière sélection : on parvient ainsi à effectuer une sélection selon 24 critères.

Classement des fiches

Il est toujours possible d'effectuer, éventuellement en même temps qu'une sélection, un classement des fiches, selon un ordre croissant ou décroissant, au choix. La principale limitation tient au fait qu'on ne peut simultanément demander une mise en réserve des numéros sélectionnés et un classement.

D'autre part, on dispose seulement de 4K en mémoire pour mémoriser les contenus des zones utilisées pour le classement, ce qui limite le classement direct à 500 fiches sur 3 caractères, ou 300 fiches sur 8 caractères, compte tenu de la place nécessaire pour mémoriser les numéros des fiches. Cette difficulté peut être tournée en opérant le classement en deux fois, chacun s'opérant sur la moitié du fichier, partitionné grâce à une sélection selon le critère de classement. L'utilisateur est prévenu par une sonnerie s'il dépasse le seuil de 4K. Dans ce cas, son classement sera incomplet, et il devra le terminer.

Les critères de sélection et, éventuellement, de classement ou de réserve, peuvent être mémorisés dans un fichier de traitement. Il est ainsi possible de définir une fois pour toutes un certain nombre de classements types qu'il suffit alors d'appeler par leur nom pour les exécuter.

Modes d'impression

Deux principaux modes d'impression sont prévus. Le premier concerne la sortie d'étiquettes, que l'on peut formater en indiquant les numéros et l'emplacement des diverses zones à sortir sur l'étiquette. L'autre mode d'impresion consiste à sortir un tableau dans lequel les zones sélectionnées pour l'impression apparaissent en colonne. Il revient à l'utilisateur de choisir les zones et la longueur réservée à l'impression pour chaque zone (on peut imprimer seulement les n premiers caractères), sachant qu'il y a une limite à 80 ou 132 colonnes, selon l'imprimante utilisée.

Les paramètres d'impression, comportant entre autres la définition des codes ASCII donnant éventuellement accès aux caractères compressés et allongés, sont placés dans un fichier se trouvant sur la disquette programme. Nous considérons qu'il s'agit là d'un choix qui n'est pas très heureux :

1) Ce fichier est unique : quand l'utilisateur a accès à deux types d'imprimantes assez différentes, il doit redéfinir tous les paramètres chaque fois qu'il change d'imprimante. C'est le problème que j'ai rencontré en voulant sortir des étiquettes sur une imprimante à aiguilles (carte parallèle), puis passer à une imprimante à marguerite (carte série) pour sortir des listes ayant une belle présentation.

2) Le fait de devoir écrire sur la disquette programme chaque fois que l'on modifie les

paramètres d'impression augmente le risque de voir cette disquette abîmée; or, comme aucune disquette de back-up n'est fournie pour le moment, cela peut poser des problèmes à l'utilisateur.

Fiches annexes

Il est possible de définir des fiches appelées "annexes" selon le mode de définition des fichiers libres. Ces fiches peuvent être fusionnées à la demande avec un fichier nominatif ou un fichier libre. Ainsi, un fichier "fournisseurs" (libre ou nominatif) pourrait être fusionné, selon le traitement que l'on désire effectuer, à un fichier annexe "articles" ou à un fichier annexe "bons de commande".

Aspects ergonomiques

CX Multigestion fonctionne à partir de menus. Le déplacement d'un écran à l'autre ou, à l'intérieur d'un écran, d'une zone à l'autre, la validation ou l'infirmation d'une information s'effectuent tous à l'aide de touches de contrôle groupées à gauche du clavier : Q, W, E, A, S, D, Z et X. Nous regrettons pour notre part que les standards implicites de nombreux progiciels ne soient pas respectés : RETURN pour valider une entrée, ESC pour l'infirmer, les flèches de gauche et de droite pour monter ou descendre d'une zone à l'autre dans l'écran.

On peut se poser aussi une question, sur le mode humoristique. Le nom de la société ayant conçu ce programme est Contrôle X, ce qui signifie pour tout Appleomane "annule ce qui précédait". Il s'agit là d'une ambition tout à fait compréhensible, mais alors pourquoi définir la touche CTRL-X dans le programme pour signifier "retour au menu précédent sans infirmation de ce qui vient de se passer" ?

Implantation en mémoire centrale

Une grande originalité de CX Multigestion tient au choix qui a été fait de mettre la quasi-totalité du programme en mémoire centrale. Ce programme comprend 15000 instructions en assembleur, les dossiers et leurs pointeurs se trouvant entièrement sur disquette. Le nombre de fiches que l'on peut mettre sur une disquette dépend bien entendu du nombre des zones et de la longueur des informations placées dans chaque zone; on peut grosso modo placer de 200 à 2000 fiches sur une disquette.

Tous les accès à la disquette ayant évidemment été écrits directement à l'aide du RWTS, les temps d'accès restent tout à fait acceptables :

. temps de recherche : environ 45 fiches par seconde
. temps de traitement, écoulé lors d'une sélection avant l'apparition du premier dossier sélectionné : 7 à 10 fiches par seconde; pour les suivants, il n'y a pas de délai sensible
. temps de classement (sans calculs) : comme le précédent, mais nécessite auparavant une lecture complète de la disquette; il faut donc environ de 12 secondes pour classer une disquette de 100 fiches et 1 minute pour une disquette de 500.

Comme vous pouvez le voir, il s'agit là de temps de réponse très satisfaisants.

La documentation

La documentation est bien présentée dans un dossier à spirales et à couverture rigide, selon la recette éprouvée par Visicalc. Elle est bien rédigée et présentée de façon agréable, mais reste un tant soit peu confuse à l'utilisation. Il arrive souvent, quand on recherche une information précise, que l'on mette un certain temps avant de trouver l'endroit où il en est question. D'autre part, une fiche cartonnée aide-mémoire (comme celle de Visicalc, par exemple) serait d'un grand intérêt - elle manque d'autant plus que le choix de touches de commande de CX Multigestion diffère des normes courantes.

Conclusion

Nous avons regretté :

. sur le plan ergonomique, le manque de cohérence des touches de fonction avec les standards implicites des progiciels courants
. sur le plan de la sécurité et de la flexibilité, le fait que les paramètres d'impression se trouvent sur la disquette programme
. au niveau de l'apprentissage, la nécessité de devoir consacrer une bonne journée à la maîtrise de l'outil
. l'aspect monolithique des entrées et sorties : pas de définition possible de masques d'entrée (format unique), et seulement deux modes possibles de sortie sur imprimante (n colonnes ou étiquettes).

Nous avons aimé :

.la rapidité de fonctionnement
.la sécurité de fonctionnement
.la facilité de mise en place d'une application, une fois le modèle maîtrisé
.la possibilité de classer ou de trier avec n'importe quelle rubrique comme critère
.la possibilité d'ajouter ou d'enlever des rubriques à un fichier existant

Nous attendons avec impatience de voir l'adaptation de ce système à des supports magnétiques de plus grande capacité et les nouveaux produits qui doivent être développés autout de CX Multigestion, ce dernier n'étant que l'axe principal d'une famille de produits.

Notre opinion générale est très favorable:

nous trouvons dans CX Multigestion un produit ayant certains points communs avec le système M/DOS. Dans les deux cas, l'utilisateur peut rapidement mettre au point des applications à base de gestion de fichiers d'une certaine complexité, et bénéficier d'un traitement efficace. La différence tient principalement à la clientèle visée, M/DOS s'adressant aux concepteurs de programmes, alors que CX Multigestion est conçu pour être utilisé par des non-informaticiens (bien qu'il vaille mieux en être un pour bien définir la structure des fiches).

NDLR : nous avons été récemment avertis que la version du programme devant sortir au SICOB a été améliorée pour tenir compte de nos remarques.

L'utilisateur pourra maintenant utiliser dans une certaine mesure des touches classiques à la place des touches de contrôle.

En outre, la nouvelle version permettra la mémorisation des paramètres d'impression sur la disquette données, au lieu de les enregistrer d'office sur la disquette du programme.

Enfin, la nouvelle documentation comportera une fiche finale aide-mémoire.

Note sur 10	0 2 4 6 8 10
Qualité de la documentation	! ********
Facilité d'apprentissage	! *******
Variété des possibilités	! **********
Rapidité de fonctionnement	! **********
Sécurité de fonctionnement	! ********
Aisance d'utilisation	! **********

Prix public de CX-Multigestion : 3.500 FF.

Courrier des lecteurs

... Je profite de cette occasion pour vous féliciter : enfin une revue spécifique de l'Apple, et en français; cela manquait cruellement. Certes, les deux premiers numéros comportaient certains articles assez "hermétiques" pour la plupart des profanes, mais heureusement les notions de base sont vite venues au secours des gens en détresse! Je dirai même que c'est grâce à votre revue que j'ai commencé à "manipuler" des programmes en langage machine et en assembleur, et que j'ai fait mes premiers pas dans ce domaine qui me paraissait bien mystérieux auparavant.

Bravo aussi pour les bancs d'essai des utilitaires et des cartes, qui permettent de se faire une idée avant d'acheter. Suggestion : vous devriez tester les différentes cartes 80 colonnes, car il y en a tellement qu'on ne sait pas laquelle choisir...

Les trucs et astuces, enfin, sont une série de trouvailles toutes plus intéressantes les unes que les autres ... que demander de plus? ... Une parution mensuelle, ou du moins bimestrielle ! Les lecteurs sont décidément insatiables ! Mais il faut reconnaître que pour les possesseurs de l'Apple ne connaissant pas l'anglais, votre revue est indispensable et irremplaçable ! Bonne continuation !

P.S. Voici une petite annonce - Vends carte SECAM 800 F (entrée par fil d'antenne TV).

Dr. Antoine Migliore - 40 Bd. Auguste Raynaud - 06100 Nice - Tel (93) 98.45.15

Votre lettre nous fait très plaisir; nous n'osons pas en général publier les lettres dithyrambiques, de peur que certains puissent croire que nous les écrivons nous-mêmes ... Mais nous n'allons pas vous priver de la publication de cette petite annonce !

Cela fait deux fois que des personnes nous promettent un article comparatif sur les cartes 80 colonnes. Nous avons préféré ne pas traiter du sujet plutôt que de le faire de façon trop cursive; la publication d'un tel article reste un de nos objectifs; nous attendons de pouvoir le faire dans de bonnes conditions.

Récent utilisateur d'un Apple II, je me

suis abonné à Pom's et je dois avouer que la majorité de son contenu passe bien au-dessus de ma tête. Cependant, avec votre aide, je ne désespère pas de faire quelques progrès dans l'avenir.

Je me permets de vous faire parvenir un court programme sans prétention qui, avec la carte RVB Chat Mauve, permet de sélectionner simplement un fonctionnement en mode monochrome parmi les 15 couleurs possibles.

```
]LIST

1   HOME
5   REM   ***PROGRAMME RVB***
6   REM
7   REM    ::: APPLE II + :::
8   REM    :::ET CARTE RVB:::
9   REM    ::: CHAT MAUVE :::
10  GR : HOME
20  FOR T = 0 TO 31: COLOR= T / 2
     : VLIN 0,39 AT T + 3: NEXT T

25  PRINT "      1   3   5   7   9
     11  13  15"
30  PRINT "        2   4   6   8
     10  12  14"
40  PRINT "COULEUR (1-15)"
45  INPUT C
50  IF C >  = 1 AND C <  = 15 THEN
     80
60  GOTO 25
80  REM  SELECTION COULEUR
85 CO = C:C1 = C:C2 = C:C3 = C:B =
     C:CT = C
90  GOSUB 2000
95  TEXT : HOME
100  HTAB 5: VTAB 2: INVERSE
110  IF C = 1 THEN  PRINT "<<<RVB
      MAGENTA>>>"
120  IF C = 2 THEN  PRINT "<<<RVB
      BLEU FONCE>>>"
130  IF C = 3 THEN  PRINT "<<<RVB
      VIOLET>>>"
140  IF C = 4 THEN  PRINT "<<<RVB
      VERT FONCE>>>"
150  IF C = 5 THEN  PRINT "<<<RVB
      GRIS 1>>>"
160  IF C = 6 THEN  PRINT "<<<RVB
      BLEU>>>"
170  IF C = 7 THEN  PRINT "<<<RVB
      BLEU CLAIR>>>"
180  IF C = 8 THEN  PRINT "<<<RVB
      MARRON>>>"
190  IF C = 9 THEN  PRINT "<<<RVB
      ORANGE>>>"
200  IF C = 10 THEN  PRINT "<<<RV
     B GRIS 2>>>"
210  IF C = 11 THEN  PRINT "<<<RV
     B ROSE>>>"
220  IF C = 12 THEN  PRINT "<<<RV
     B VERT CLAIR>>>"
230  IF C = 13 THEN  PRINT "<<<RV
     B JAUNE>>>"
240  IF C = 14 THEN  PRINT "<<<RV
     B TURQUOISE>>>"
250  IF C = 15 THEN  PRINT "<<<RV
     B BLANC>>>": POKE  - 16144,0
300  NORMAL
310  END
2000  POKE  - 16139,16 * (15 - CT
     )
2010  POKE  - 14577 - B,15 - C0
2020  POKE  - 14561 - B,15 - C1
2030  POKE  - 14545 - B,15 - C2
2040  POKE  - 14529 - B,15 - C3
2050  POKE  - 16131,16 * (15 - CT
     )
2060  RETURN
```

Dr. J-F Schwartz - 54300 Lunéville

Nous sommes bien conscients que certains articles sont d'un niveau difficilement accessible au néophyte. Dans la mesure où nous nous adressons à tous les utilisateurs francophones d'Apple, nous sommes en effet tenus de panacher le niveau des articles afin que chacun y trouve à boire et à manger. Nous n'avons pas eu la place de mettre dans ce numéro un article de la série sur les notions de base : cela reprendra cependant dès le prochain numéro, ainsi que la série sur l'apprentissage de l'assembleur, si elle nous parvient à temps ...

L'imprimante étant absente de mon système micro-informatique, je prends la liberté de vous écrire à la main. D'abord : BRAVO ! Oui, je tenais à vous féliciter pour l'idée, la réalisation et la parution de votre revue. Vous pouvez croire que si un flemmard du stylo comme moi prend la peine de vous l'écrire, c'est que vous l'avez vraiment mérité.

J'ai pris connaissance de Pom's depuis peu, et je me suis régalé. Voici enfin la lecture saine, instructive et exclusive que ne pouvaient qu'espérer tous les passionnés de l'Apple. Je m'initie à cette bête de machine depuis près d'un an, et j'ai l'impression d'avoir encore pas mal à en tirer.

Brièvement, ma petite critique de Pom's.

- *les points positifs :*
  - *la présentation : +++*
  - *les programmes : ++++ (par leur choix, leur essence et leur composition*
  - *le titre : +++ (peut-être pas original, mais évident !)*
  - *la lisibilité des listings : +++++ (sauf pour la lettre D dans les listes en hexa qui ressort comme un C)*
- *les points négatifs (eh oui)*
  - *la longueur de la revue, par conséquent son épaisseur (on reste sur sa faim)*
  - *les intervalles de parution (trop longs)*
  - *les explications de ce que fait le programme, parfois assez succintes.*

Pour conclure, je reviendrai sur les programmes en Applesoft, qui pour certains sont composés de manière plus que claire, et surtout astucieuse. J'apprécie beaucoup les

logiciels faisant preuve d'astuce de la part du programmeur. Ces gens-là sont-ils des professionnels, ou de simples amateurs comme moi ?

Dimitri Auer - 68130 Wittersdorf

Nous avons déjà résolu en partie le problème de l'importance de la revue depuis le numéro 4 : nous réduisons maintenant le texte, ce qui nous a fait gagner l'équivalent de 10 pages de texte. Nous ne pouvons hélas pas encore augmenter la fréquence; il nous faut pour cela passer le cap des 2000 abonnés.

Les auteurs de tous les articles des deux premiers numéros étaient des professionnels. Nous recevons de plus en plus de contributions, ce qui fait que plusieurs de nos auteurs actuels sont des amateurs; c'est en particulier le cas d'Olivier Herz à qui nous devons HAIFA et de Denis Sureau (le clavier magique et la programmation facilitée).

Bibliographie

1. Les éditions Radio viennent de sortir plusieurs ouvrages intéressants et présentés de façon agréable :

Initiation BASIC, de H. Lilen, 176 pages - D'intérêt général, mais avec les exemples en BASIC Microsoft.

Pratique de l'Apple II, de H. Lilen, 192 pages - 80 FF - Très clair.

70 programmes BASIC, de Lon Poole, Mary Borchers et David Castlewitz, 198 pages - 85 FF - Pourquoi ne pas fournir la disquette en option ?

Manuel de l'utilisateur Apple II, de Lon Poole, Martin McNiff et Steven Cook, 367 pages - 95 FF - Assez complet, mais moins distrayant à lire.

2. Les éditions du PSI vous offrent deux nouveaux titres :

La comptabilité sur Apple, de S. et G. Llilio - 95 FF - Sort au SICOB.

La pratique du Visicalc sur Apple, de Hervé Thiriez - Prix et nombre de pages inconnus au moment de la rédaction de Pom's - Comporte de nombreux exemples d'applications, allant de l'initiation à Visicalc aux astuces professionnelles de création de tableaux. Sort au SICOB.

pom's

BULLETIN D'ABONNEMENT

Je désire recevoir le N° ___ de POM'S

☐ avec disquette _____ 85 F TTC

☐ sans disquette _____ 35 F TTC

Je désire m'abonner pour 4 numéros

à partir du N° ___

☐ avec disquette _____ 295 F TTC

☐ sans disquette _____ 120 F TTC

Nom _____

Adresse _____

Ces tarifs comprennent l'envoi postal en France Métropolitaine et CEE (voie aérienne exceptée)

Envoyez ce bon de commande et votre règlement à :

Editions MEV - 49 rue Lamartine - 78000 Versailles

Imprimerie Laballery, 58500 Clamecy, imprimé en France. Directeur de la publication Hervé Thiriez
Dépôt légal : mai 1983

PRENEZ UN AN D'AVANCE EN MICRO INFORMATIQUE !

SICOB BOUTIQUE

CNIT - PARIS LA DÉFENSE
DU 22 SEPT. AU 1er OCT. 82
DE 9 H 30 A 18 H - FERMÉ LE DIMANCHE 26
ENTRÉE LIBRE

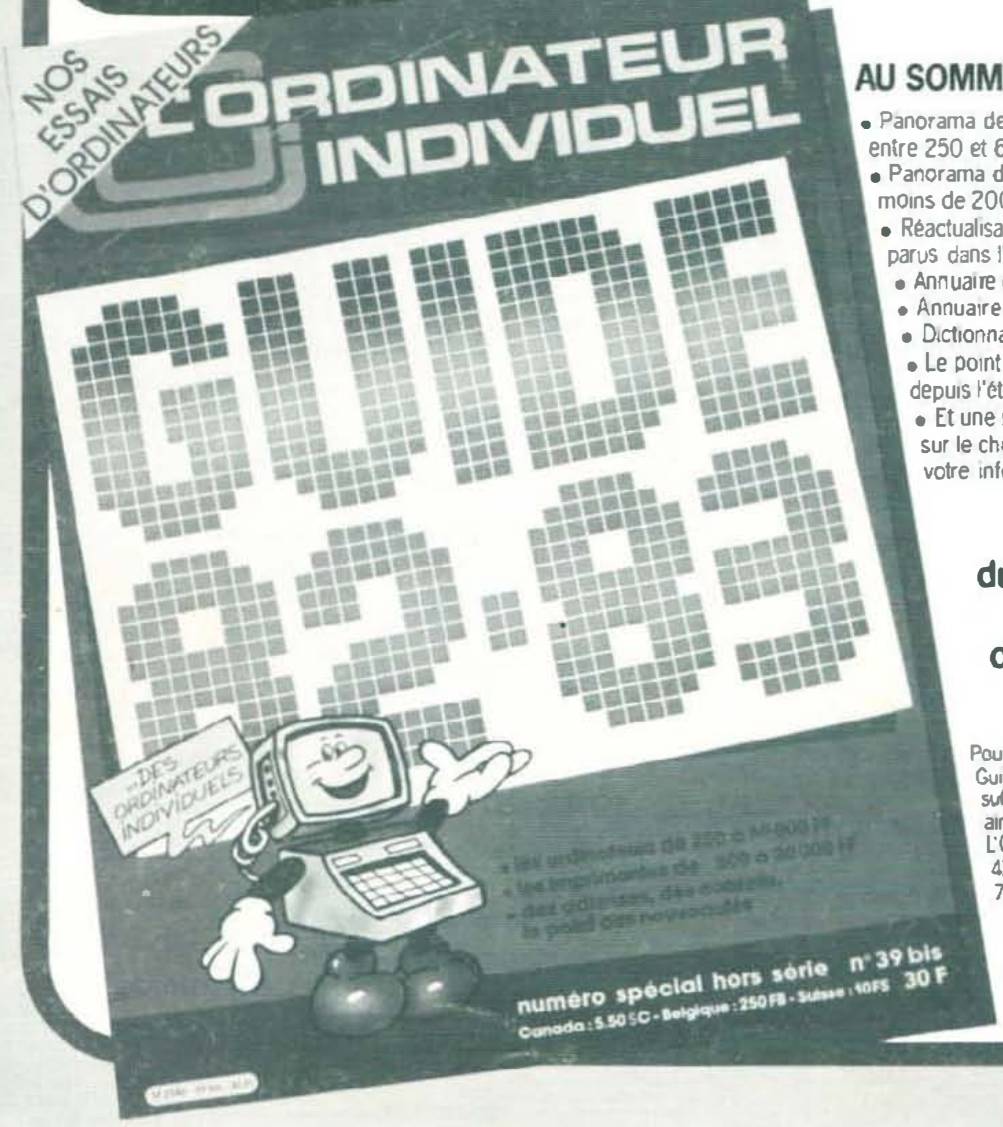
V 8302

PUBLICIS

UN DOCUMENT ESSENTIEL

GUIDE 82-83

DE L'ORDINATEUR INDIVIDUEL



AU SOMMAIRE

- Panorama des ordinateurs valant entre 250 et 60000 FF (plus de 120 matériels).
- Panorama des imprimantes valant moins de 20000 FF (plus de 70 matériels).
- Réactualisation de 12 bancs d'essai parus dans l'OI.
- Annuaire des fournisseurs (plus de 700 adresses).
- Annuaire des clubs (plus de 200 adresses).
- Dictionnaire de l'informatique individuelle.
- Le point sur les nouveautés parues depuis l'été 1981.
- Et une série d'articles pour vous "guider" sur le chemin de votre informatisation individuelle.

à partir du 10 septembre 1982
30 FF
chez votre marchand de journaux

Pour recevoir, chez vous le Guide 82 83 dès sa parution, il vous suffit d'envoyer vos nom et adresse ainsi qu'un chèque de 30 FF à L'ORDINATEUR INDIVIDUEL (GUIDE 82-83) 41 rue de la Grange-aux-Belles 75483 Paris Cedex 10

Une réduction de 5 FF est accordée aux abonnés sur envoi de la dernière étiquette d'expédition

Erratum du N° 5 (paru dans le N° 6)

1. Le clavier magique

Page 4, colonne 2, ligne 4, il faut lire : #<DEPART >, <INCREMENT >.

2. La programmation facilitée

Page 23, colonne 2, lignes 6 et 7, lire : X < 128 diminue la durée, X > 128 augmente.

15 lignes plus bas, lire « affichant PRINT FN ARR(X) »

Afin de formater un écran entier avant de placer les INPUT et de créer une commande de formatage pur, il suffit de quelques instructions en plus et de créer une commande de formatage pur. On supprime « INPUT A$ » et le paramètre longueur devient obligatoire, car le « 1 » servira à identifier la commande. Il faut pour cela :

— Modifier les lignes suivantes :

```
 72 LDY #·0
 73 STY LONGUEUR
 74 STY $6
123 BEQ MASQUE
```

— Insérer les lignes suivantes :

```
 54 CMP #'1'
 55 BEQ INPUT

 75 PHA
 76 CMP #'1'
 77 BEQ INPTL

115 DEC $BF

122 INC $BF

137 MASQUE PLA
138        CMP '1'
139        BEQ GTRTS
```

3. HAIFA

Page 35, col. 2, § 3, ligne 5, lire « #$A5 ».

Paragraphe suivant, lignes 4 et 5, lire « fait un bip, déconnecte les routines spéciales d'entrée/sortie, puis la reprogrammation de HAIFA intervient — le RESET agit comme CTRL-C : il fait".

Page 36, la pile. Lire PUSH > et PULL > (pas de signe "–").

Page 36, à propos de & GOTO, & GOSUB, & RESTORE et & NEW : comme on peut utiliser des expressions numériques à la place des nombres, il vaut mieux ne plus utiliser le RENUMBER !

Page 37, le « INPUT anything » Une propriété supplémentaire a été omise : & INPUT « [chaîne] » ; [nom de variable], [exp. num.]
& INPUT [nom de variable], [exp. num.].

Ces commandes agissent comme & INPUT, mais ajoutent derrière le curseur un nombre de points correspondant à l'expression numérique (de 1 à 255). Ce nombre de points indique la longueur maximale de la chaîne autorisée : en cas de dépassement, la chaîne sera tronquée en conséquence.

Page 42, bas de colonne 1 : le type 9 a pour effet « T<=>E ».

Page 43, col. 1, §6, lignes 6 et 7 : ...« < » pour la gauche, « > » pour la droite...

Page 43, tracer des points et des droites : le HPLOT, le X HPLOT et le N HPLOT peuvent être prolongés par chacune des continuations [ex.1], [ex.2] ou TO [ex.1], [ex.2] ou ON [ex.3].

Page 45, col. 2, §4, lire PR# et IN# au lieu de PR= et IN=.

Page 46, §2 et page 47, ligne 1, lire CTRL-SHIFT-N au lieu de CTRL-%.

Page 46, dernier §, lire CTRL-SHIFT-P au lieu de CTRL-à.
Adresse des routines : GETLIN ($FD6A), RDKEY($FDOC) et COUT1($FDFO).

Page 48, col. 2, §4, ligne 1 : « Cette routine court-circuite l'entrée de » : ligne 9 : lire PRINT CHR$(4)« IN# 0 » : « IN » et non « PR ».

Page 51, col. 2, §2 : ligne 3 : « ...tables après les fichiers » :, ligne 9 : « du code (HAIFA.CODE en 48K ou HAIFA. CODE 1 en 64K), il ne ».

Remarques sur la disquette Pom's 5.

Le programme source est constitué de deux fichiers LISA 2.5 version 64K enchaînés par ICL car l'ensemble était trop gros pour tenir en mémoire en un seul fichier. L'assemblage de ces fichiers sauve automatiquement les deux fichiers source. Là encore, il fallait deux fichiers distincts car LISA ne laisse que 4K pour le fichier généré.

L'assemblage des deux versions de HAIFA (48K et 64K) a été réalisé avec LISA en version 64K. Pour pouvoir lister les fichiers avec LISA 48K, il a fallu découper HAIFA.TEXT 1 et HAIFA. TEXT 2 en deux parties chacun (suffixes A à D). Ces quatre fichiers s'enchaînent avec ICL, mais ne sont pas assemblables en 48K car la table des symboles est saturée.

Celle-ci a en effet une longueur de $800 en LISA 48K et de $1500 en LISA 64K. HAIFA est donc beaucoup trop gros pour être assemblé en 48K !

4. Conversion Pascal/ BASIC/Pascal

Nous adressons nos excuses aux lecteurs ayant essuyé des difficultés lors de l'utilisation des programmes de conversion Pascal/BASIC/Pascal, fournis avec les disquettes Pom's 2 à 4, et commentés dans le numéro 3.

Pour ceux d'entre vous qui reçoivent les disquettes, utilisez le programme de transfert de BASIC vers Pascal de la disquette Pom's 5, dans laquelle le bug avait été corrigé, ou celui de la disquette 6, bien entendu.

Ce bug avait été malheureusement absent de nos essais, dans lesquels nous avions remis les programmes sur une disquete Pascal vierge. La conséquence du bug est qu'une partie du programme Pascal risque d'être absente en fin de course.

Le remède est le suivant, à appliquer aux programmes présentés par Gilles Mauffrey dans le Pom's 3 :

1. Dans PASCBAS.OBJO, ligne 288, lire CMP # $10.

Donc, dans le code objet, faire AE3 : 10.

2. Dans BASPASC.OBJO (sur disquette uniquement), ligne 449, lire CMP # $10.

Donc, dans le code objet, faire C29 : 10.

3. Avant toute utilisation, nettoyer la mémoire avec :

2000 : 0
2001 < 2000 . 8000 M.